Telemetry Ranging: Signal Processing

Jon Hamkins*, Peter Kinman†, Hua Xie*, Victor Vilnrotter*, and Sam Dolinar*

ABSTRACT. — This article describes the details of the signal processing used in a telemetry ranging system in which timing information is extracted from the downlink telemetry signal in order to compute spacecraft range. A previous article describes telemetry ranging concepts and architecture, which are a slight variation of a scheme published earlier. As in that earlier work, the telemetry ranging concept eliminates the need for a dedicated downlink ranging signal to communicate the necessary timing information.

The present article describes the operation and performance of the major receiver functions on the spacecraft and the ground – many of which are standard tracking loops already in use in JPL's flight and ground radios – and how they can be used to provide the relevant information for making a range measurement. It also describes the implementation of these functions in software, and performance of an end-to-end software simulation of the telemetry ranging system.

I. Introduction

A previous article [1] describes the basic concepts behind both telemetry ranging and conventional two-way ranging. In brief, a signal is sent from a ground station to the spacecraft, which reacts and sends a return signal to Earth. For a return signal arriving at the ground station at time t_R, we can back-calculate the time t_T associated with the originating ground transmission, which provides the two-way time delay $t_R - t_T$. Mathematically, t_T is determined by solving for the lower limit of an integral in a

*Communications Architectures and Research Section.

†California State University, Fresno.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.
© 2016 All rights reserved.

canonical range equation determined from Equations (3), (4), and (5) of [1]:

$$\int_{t_T}^{t_R} \dot{\psi}_T(t)\,dt = \psi_T(t_R) - \psi_T(t_T) \tag{1}$$

$$= \begin{cases} \psi_T(t_R) - \psi_R(t_R), & \text{conventional two-way ranging} \\ \psi_T(t_R) - \psi_S(t_S), & \text{telemetry ranging.} \end{cases} \tag{2}$$

In (1), all values are known except t_T: $\psi_T(t_R)$ is the phase of the uplink transmission sampled at a known time t_R on the ground; $\psi_T(t_T)$ is the phase of the same uplink transmission recorded and retrievable from Deep Space Station (DSS) records at the earlier unknown time t_T; and a record of the derivative of the transmitted phase $\left\{\dot{\psi}_T(t), t \in [t_T, t_R]\right\}$ is assumed to be available back to the departure time t_T of the corresponding uplink signal. The two-way time delay computed from (2) includes some processing delays that must be removed by calibration. See the previous article [1] for details on relating the time delay $t_R - t_T$ calculated from (2) to the two-way geometric delay of interest.

Conventional two-way ranging and telemetry ranging use different methods to infer the value of the transmitted phase $\psi_T(t_T)$ present at the DSS transmitter at the (initially) unknown time t_T. In conventional ranging, this phase is inferred by measuring the received phase $\psi_R(t_R)$ when the return signal from the spacecraft arrives at the DSS receiver. In telemetry ranging, this same phase is inferred by measuring a phase $\psi_S(t_S)$ at the spacecraft at an unknown time t_S but associated with a telemetry frame observed later at the DSS receiver at time t_R. These two alternative methods for evaluating $\psi_T(t_T)$ are shown in (2).

In this article we describe, for telemetry ranging only, the signal processing steps performed at the DSS and on the spacecraft to obtain the two phases $\psi_T(t_R)$, $\psi_S(t_S)$ needed in (2), without having explicit knowledge of the transmitter time t_T or the spacecraft time t_S corresponding to a given arrival time at the receiver t_R. In Section II, we describe the mathematical form of the uplink and downlink signals used by a telemetry ranging system. Sections III and IV describe the signal processing at the spacecraft receiver and DSS receiver, respectively. These signal models form the basis for our detailed description of the uplink and downlink signal processing steps in the remainder of this article. Sections V and VI describe a software implementation of the signal processing, and Section VII gives performance analysis and numerical results.

II. Signal Model

A. Uplink PN Range Signal

A Pseudo-random Noise (PN) ranging signal is formed by modulating a range code, or sequence, using pulse shaping and a residual carrier. The signal has the form

$$s(t) = \sqrt{2P_t}\cos[2\pi f_c t + \phi_r \phi_{\text{PN}}(t)], \tag{3}$$

where

$$P_t \text{ is the total signal power} \tag{4}$$

$$f_c \text{ is the carrier frequency in Hz} \tag{5}$$

$$\phi_r \text{ is the peak modulation index in radians,} \tag{6}$$

and where $\phi_{\rm PN}(t)$ is the PN waveform we shall describe next. A functional block diagram of the uplink signal generation is shown in Figure 1.

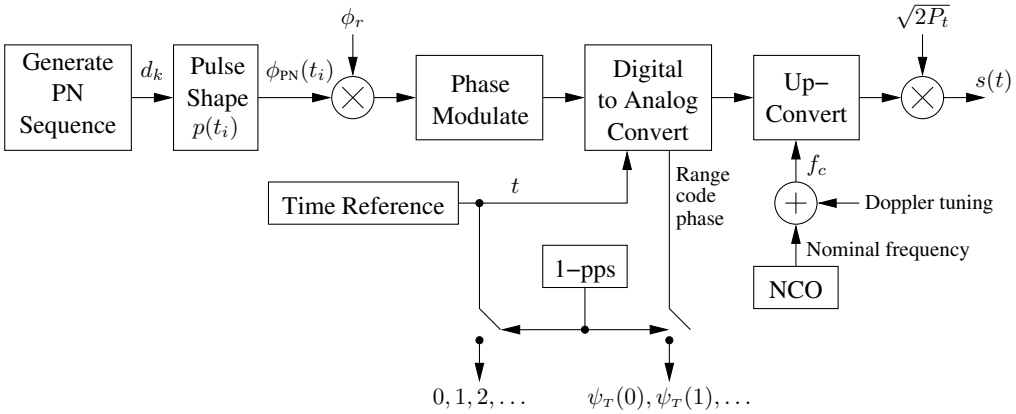


Figure 1. Generation of the uplink ranging signal $s(t)$ and uplink ranging statistics t_T and $\psi_T(t_T)$.

1. The PN Sequence

The basic building block of the ranging signal is a PN sequence, or chip sequence, $\{d_k\}$, shown in the leftmost block of Figure 1. In this section, we describe three closely related chip sequences. One of them is supported by the Deep Space Network (DSN) [2] and was used on the New Horizons mission [3], and the other two are CCSDS standards [4]. All three are composite codes built from the six component codes shown in Table 1. Each component code is periodic, so that the kth output of the jth code is

$$C_j(k) = C_j(k \mod L_j), \tag{7}$$

where L_j is the period of the jth code. $C_1(\cdot)$ simply toggles between 1 and 0 and is called the *range clock*. The periods of the component codes are 2, 7, 11, 15, 19, and 23. The component codes have historically been referred to as PN codes, giving rise to the name "PN ranging" for a ranging system that uses them. We follow this well-accepted usage from the literature, notwithstanding that a reasonable argument could be made that the component codes are not PN codes, because most of them are not the output of a maximum length shift register, and their spectrum does not look like that of random data.

Table 1. Component codes.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
C_1	1	0																					
C_2	1	1	1	0	0	1	0																
C_3	1	1	1	0	0	0	1	0	1	1	0												
C_4	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0								
C_5	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	1	1	0	0				
C_6	1	1	1	1	1	0	1	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0

The three PN range codes are given by:

$$d_{\text{DSN}}(k) = \{C_1(k) \text{ OR } [C_2(k) \text{ AND } C_3(k) \text{ AND } C_4(k) \text{ AND } C_5(k) \text{ AND } C_6(k)]\}' \quad (8)$$

$$d_{\text{T2B}}(k) = \text{sgn}\left(2C_1'(k) + C_2'(k) - C_3'(k) - C_4'(k) + C_5'(k) - C_6'(k)\right) \quad (9)$$

$$d_{\text{T4B}}(k) = \text{sgn}\left(4C_1'(k) + C_2'(k) - C_3'(k) - C_4'(k) + C_5'(k) - C_6'(k)\right), \quad (10)$$

where in this context, prime notation converts a binary-coded sequence of 0s and 1s into an equivalent binary sequence of +1s and −1s:

$$c' = \begin{cases} 1, & \text{if } c = 1 \\ -1, & \text{if } c = 0. \end{cases} \quad (11)$$

The PN range code in (8) is used in the DSN [2]. The PN range code in (9) is the weighted-voting balanced Tausworthe code with $v = 2$, known as T2B [4]. The PN range code in (10) is the weighted-voting balanced Tausworthe code with $v = 4$, known as T4B [4]. The constructions of the T2B and T4B codes differ only in the number of "votes" that the range clock $C_1'(\cdot)$ is given.

Since the periods of the component codes are relatively prime, each of the DSN, T2B, and T4B range codes has period

$$L = \text{lcm}_{j:1\le j\le 6}\{L_j\} = \prod_{j=1}^{6} L_j = 2 \times 7 \times 11 \times 15 \times 19 \times 23 = 1,009,470. \quad (12)$$

In the DSN code, the 5-argument logical AND in the square bracket term of (8) is mostly logically false, so that the DSN range code is highly correlated with the range clock. Similarly, the extra range clock voting power in T2B and T4B produces a high correlation with the range clock. For example, the T4B sequence defined in (10) begins

$$1, -1, 1, -1, 1, 1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1. \quad (13)$$

A positive correlation with the C_1' range clock is readily apparent: among the first 20 sequence values shown, 19 agree with C_1'.

2. Pulse Shaping

The PN sequence of +1s and −1s is modulated within the duration of one chip time T by a pulse shape $p(t)$. This produces a periodic waveform $\phi_{\text{PN}}(t)$, with period 1,009,470 chip times, defined by

$$\phi_{\text{PN}}(t) = \sum_{k=-\infty}^{\infty} d_k p(t - kT), \tag{14}$$

where

$$d_k \in \{-1, +1\} \text{ is the range code (PN sequence)} \tag{15}$$

$$T \text{ is the chip duration} \tag{16}$$

$$p(t) \text{ is the pulse shape defined over the interval } [0, T). \tag{17}$$

The range signaling established for the DSN [2] and in the international standards [4] permits either a rectangular pulse shape,

$$p(t) = \begin{cases} 1, & t \in [0, T) \\ 0, & \text{else,} \end{cases} \tag{18}$$

when the chip rate is low, or a positive half-sine pulse shape,

$$p(t) = \begin{cases} \sin(\pi t/T), & t \in [0, T) \\ 0, & \text{else,} \end{cases} \tag{19}$$

for typical higher chip rates. The waveform $\phi_{\text{PN}}(t)$ resulting when the T4B sequence is used with either a rectangular or a half-sine pulse shape is shown in Figure 2. The values of the T4B sequence can be seen as positive and negative regions of the $\phi_{\text{PN}}(t)$ waveform when the rectangular pulse shape is used. With the half-sine pulse shape, a pure sine wave results in those regions that match C_1'.

By comparison, the "range clock waveform" $\phi_{\text{RC}}(t)$ that would result from directly modulating the range clock sequence $C_1'(k)$ is

$$\phi_{\text{RC}}(t) = \begin{cases} \text{sgn}(\sin(2\pi f_{\text{RC}} t)), & p(t) \text{ is rectangular shape} \\ \sin(2\pi f_{\text{RC}} t), & p(t) \text{ is half-sine shape,} \end{cases} \tag{20}$$

where $f_{\text{RC}} \triangleq 1/(2T)$ is the frequency of the range clock waveform (i.e., square wave or sine wave) resulting from the C_1 clock sequence. Note that the period of the range clock waveform $\phi_{\text{RC}}(t)$ is two chip durations, i.e., $2T$. For each of the three sequences we defined, the PN range code waveform $\phi_{\text{PN}}(t)$ is well correlated with the pure range clock waveform $\phi_{\text{RC}}(t)$. For the T4B sequence used as the example in Figure 2, the correlation coefficient is approximately 0.93.

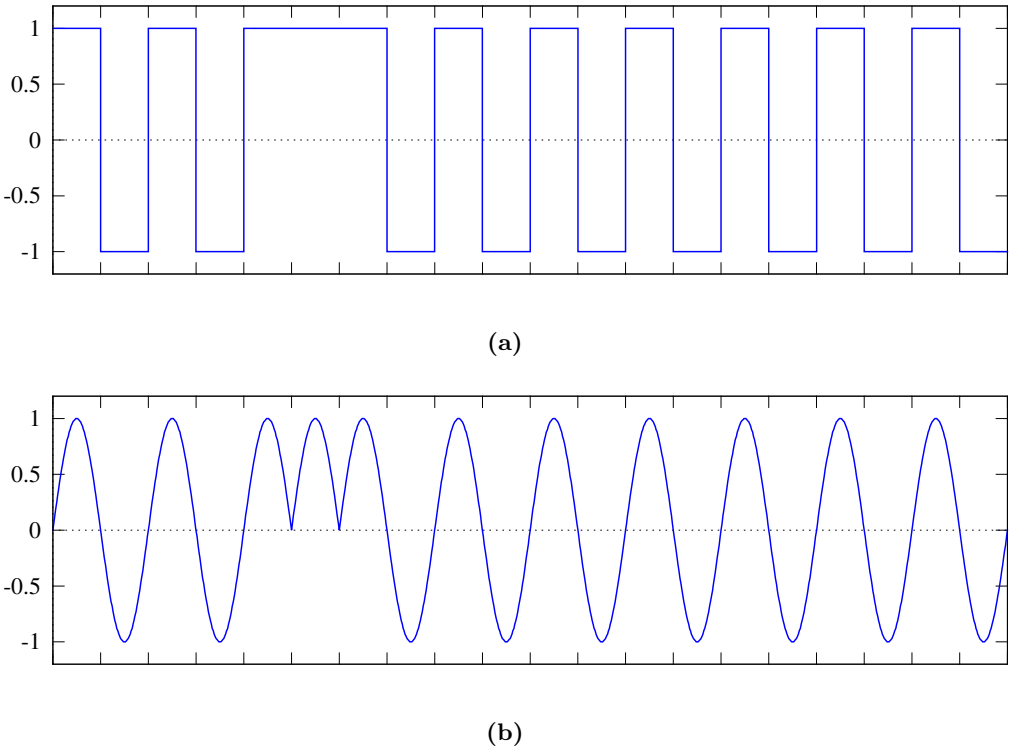


Figure 2. The waveform $\phi_{\rm PN}(t)$ resulting when the T4B sequence is used with either (a) a rectangular pulse shape or (b) a half-sine pulse shape.

3. Digital to Analog Conversion

In a modern implementation, most of the processing is done digitally in discrete time. The sample times may be denoted $t_i \triangleq iT_s$, as indicated in Figure 1, where T_s is a sample period. In this way, the blocks preceding the digital-to-analog converter (DAC) are not time-sensitive, and indeed they may be pre-computed and stored in a buffer. The DAC takes discrete samples of the form $\cos(\phi_r \phi_{\rm PN}(t_i))$ and produces a continuous-time signal of the form $\cos(\phi_r \phi_{\rm PN}(t))$. At one second intervals, controlled by a one pulse-per-second (1-pps) signal, the time t and associated range code phase $\psi_T(t)$ being processed by the DAC are recorded at a ground station transmission reference point $\mathcal{P}_{\rm T}$.

The transmitted uplink signal is customarily compensated for Doppler, using predicts developed from ephemerides obtained via navigation solutions of previous ranging tracks. This is done by adjusting f_c in (3) higher or lower, as shown in Figure 1, introducing some time-dependence on the uplink frequency.

B. Downlink Telemetry Signal

The role of the downlink in a telemetry ranging system is to convey the phase $\psi_S(t_S)$ tracked at the spacecraft back to the DSS as part of the normal downlink telemetry stream. Analogously to the uplink signal, the downlink signal is generated from a

binary sequence $\{d_k\}$ of telemetry data that is pulse-shaped, phase-modulated, and up-converted. The form of the signal is

$$s(t) = \sqrt{2P_t}\cos[2\pi f_c t + \phi_d \phi_{\text{TM}}(t)], \tag{21}$$

where

$$P_t \text{ is the total signal power} \tag{22}$$

$$f_c \text{ is the downlink carrier frequency in Hz} \tag{23}$$

$$\phi_d \in (0, \pi/2] \text{ is the telemetry data modulation index in radians} \tag{24}$$

$$\phi_{\text{TM}}(t) = \sum_{i=-\infty}^{\infty} d_k p(t - iT) \tag{25}$$

$$d_k \in \{-1, +1\} \text{ is the downlink telemetry data} \tag{26}$$

$$T \text{ is the symbol duration} \tag{27}$$

$$p(t) \text{ is a pulse-shape defined over the interval } [0, T). \tag{28}$$

The spacecraft receiver's range code phase measurements are encoded within the downlink telemetry data $\{d_k\}$. A functional block diagram of this is shown in Figure 3. In a slight abuse of notation, we re-use P_t, f_c, d_k, T, and $p(t)$ for describing the downlink signaling. It should be understood that these parameters have the same meanings as in the uplink signal, but in general they will not have the same values used in the uplink signal.

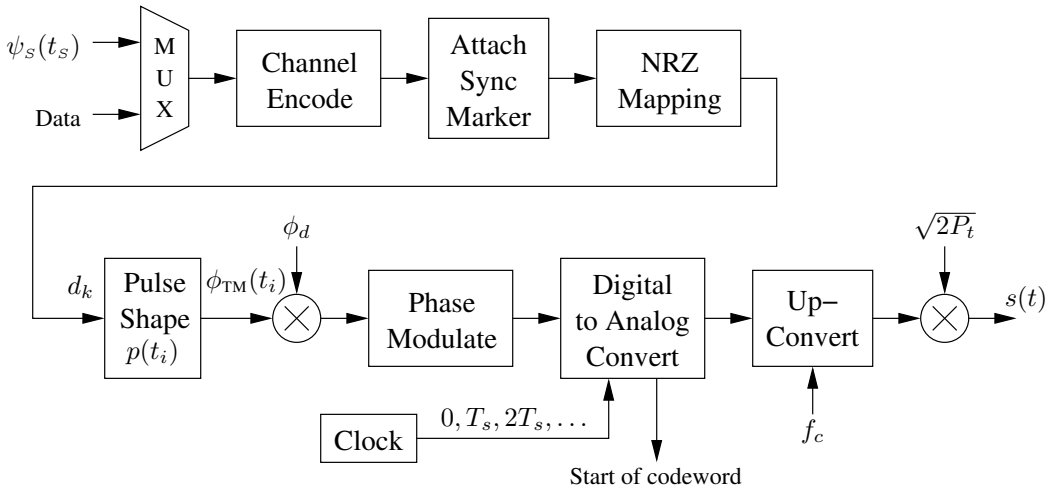


Figure 3. Generation of the downlink telemetry signal $s(t)$.

Unlike the generation of the uplink signal, the telemetry ranging concept does not require an accurate spacecraft clock or the ability to accurately measure delays on board. The clock shown in Figure 3 may be slow or have high Allan variance. What is important, however, is the ability to generate a "start of codeword" signal that corresponds with the time the DAC produces the first symbol of a codeword, at the spacecraft reference

point $\mathcal{P}_S$. This signal, occurring at time t_S, controls when to latch (record) the spacecraft range code phase measurement $\psi_S(t_S)$.

III. Signal Processing in the Uplink Receiver

The range code phase measurement $\psi_S(t_S)$ needed in (2) is carried out by tracking the uplink signal. This section describes the signal processing used in the spacecraft receiver to track the uplink carrier phase, fractional PN chip timing, and range code phase.

The spacecraft receiver down-converts, digitizes, and tracks the uplink signal. A functional block diagram of the spacecraft receiver is shown in Figure 4. For the moment, we do not consider analog-to-digital conversion and discuss each of the signal processing steps in the figure in the analytically convenient continuous-time basis. Section V will remove this assumption and describe the discrete-time implementation.

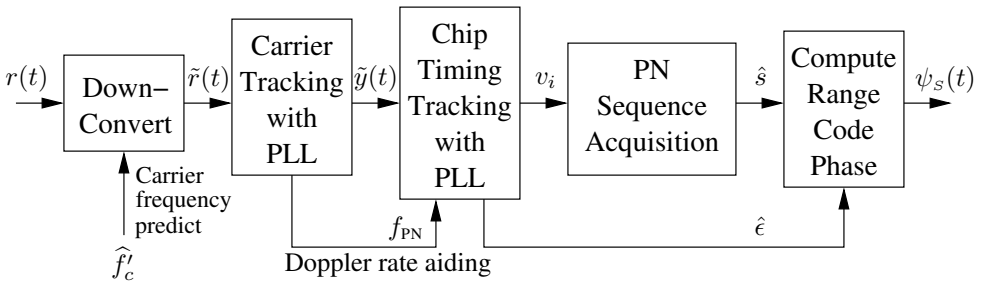


Figure 4. The uplink receiver on board the spacecraft.

We will step through each of the signal processing functions shown in Figure 4. To begin, the spacecraft receives a noisy, delayed version of (3), which can be represented as

$$r(t) = s'(t) + n(t), \tag{29}$$

where $s'(t)$ is a delayed version of the transmitted signal $s(t)$, and $n(t)$ is a white Gaussian noise process with two-sided spectral density $N_0/2$. The received signal is filtered, so that $n(t)$ may be taken to be a bandpass process with zero power at frequencies outside of the signal band.

In (29), the argument t is the local spacecraft time reference. The spacecraft reference may have an unknown offset, drift, or instability with respect to the ground time reference, and so the spacecraft makes no attempt to measure a one-way delay based on its observation of $r(t)$. Instead, the job of the spacecraft signal processing is simply to track the phase of the received range code waveform $\phi'_{\text{PN}}(t)$ present within $s'(t)$ as it arrives. This tracked phase can be latched at any moment triggered by other spacecraft events.

The uplink delay may be time-varying due to the dynamically changing spacecraft range. This introduces a Doppler shift in the received carrier frequency, as well as a delay in

the time of the arriving chips. As a result, the received signal can be written [5]

$$\begin{aligned} r(t) &= s'(t) + n(t) \\ &= \sqrt{2P_t}\cos[j2\pi f_c' t + \theta_0 + \phi_r \phi'_{\text{PN}}(t)] + n(t), \end{aligned} \tag{30}$$

where f_c' is the Doppler-shifted received carrier frequency, θ_0 is an unknown phase offset, and $\phi'_{\text{PN}}(t)$ is the PN range code waveform as received at the spacecraft. In general, the Doppler-shifted frequency f_c' can itself be time-varying, but this is a second-order effect that can usually be ignored. For example, if the spacecraft is moving away from the DSS at constant non-relativistic velocity v, then $f_c' = f_c(1 - v/c)$.

A. Down-Conversion to Baseband

The received signal $r(t)$ is down-converted to baseband in the usual way, by mixing it with a local oscillator, as shown in Figure 5. The local oscillator frequency is $\widehat{f}_c'$, which is the receiver's estimate of the arriving frequency f_c'.

The down-conversion results in the quadrature components [6]

$$r_I(t) \triangleq \text{LPF}\left[r(t)\sqrt{2}\cos\left(2\pi \widehat{f}_c' t\right)\right] = \sqrt{P_t}\cos(\phi_r \phi'_{\text{PN}}(t) + \theta(t)) + n_c(t) \tag{31}$$

$$r_Q(t) \triangleq \text{LPF}\left[-r(t)\sqrt{2}\sin\left(2\pi \widehat{f}_c' t\right)\right] = \sqrt{P_t}\sin(\phi_r \phi'_{\text{PN}}(t) + \theta(t)) + n_s(t), \tag{32}$$

where

$$\theta(t) = \theta_0 + 2\pi\left(f_c' - \widehat{f}_c'\right)t. \tag{33}$$

For operational spacecraft orbiting Mars or in cruise mode, the ephemerides are typically accurate, and thus $\widehat{f}_c'$ is sufficiently close to f_c' that the residual error $\theta(t)$ is a slowly varying phase, typically on the order of 10^{-3} Hz, implying a change of one cycle or 2π radians over 1,000 seconds.

In (31) and (32), $n_c(t)$ and $n_s(t)$ are each low-pass white Gaussian noise processes with two-sided power spectral density $N_0/2$ within the signal band. The low-pass filter has

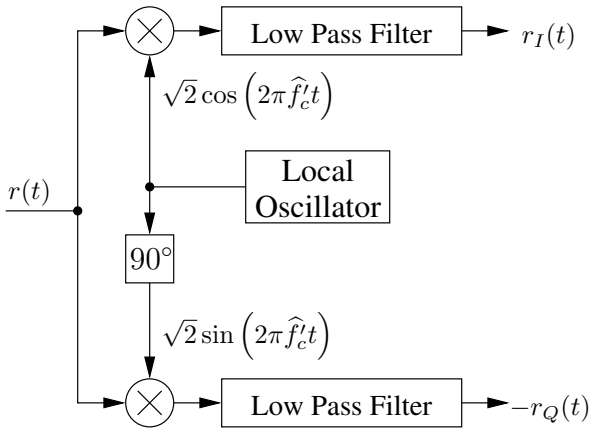


Figure 5. The received signal is down-converted.

eliminated the double frequency terms from (31) and (32). Thus, the complex baseband representation of the received signal is

$$\tilde{r}(t) \triangleq r_I(t) + jr_Q(t) = \underbrace{\sqrt{P_t}\exp[j(\phi_r\phi'_{\text{PN}}(t) + \theta(t))]}_{\triangleq\, \tilde{s}(t)} + \tilde{n}(t), \tag{34}$$

where $\tilde{s}(t)$ is the complex baseband representation of the noise-free signal and $\tilde{n}(t) \triangleq n_c(t) + jn_s(t)$ is the low-pass white noise. One can relate the bandpass representation to the complex baseband $\tilde{r}(t)$ by

$$r(t) = \sqrt{2}\,\text{Re}\left[\tilde{r}(t)e^{j2\pi\widehat{f}'_c t}\right], \tag{35}$$

which can be verified by plugging (34) into (35), comparing to (30), and noting that the bandpass noise has quadrature representation [7]

$$n(t) = \sqrt{2}n_c(t)\cos\left(2\pi\widehat{f}'_c t\right) - \sqrt{2}n_s(t)\sin\left(2\pi\widehat{f}'_c t\right). \tag{36}$$

We may rewrite (34) as

$$\tilde{r}(t) = \sqrt{P_t}e^{j\theta(t)}\left[\cos(\phi_r\phi'_{\text{PN}}(t)) + j\sin(\phi'_{\text{PN}}(t)\phi_r)\right] + \tilde{n}(t). \tag{37}$$

When $p(t)$ is the rectangular pulse shape, we have $\phi'_{\text{PN}}(t) \in \{-1, +1\}$, and so (37) becomes

$$\begin{aligned}\tilde{r}(t) &= \sqrt{P_t}e^{j\theta(t)}\left[\cos(\phi_r) + j\phi'_{\text{PN}}(t)\sin(\phi_r)\right] + \tilde{n}(t) &(38)\\ &= \sqrt{P_c}e^{j\theta(t)} + j\sqrt{P_r}e^{j\theta(t)}\phi'_{\text{PN}}(t) + \tilde{n}(t), &(39)\end{aligned}$$

where the carrier power and ranging signal power are, respectively,

$$\begin{aligned}P_c &= P_t\cos^2(\phi_r) &(40)\\ P_r &= P_t\sin^2(\phi_r), &(41)\end{aligned}$$

and the total power is $P_t = P_c + P_r$. For example, when $\phi_r = 1.23$, $P_t = 1$, $\theta(t) = 0$, $N_0 = 0$, and $p(t)$ is as defined in (18), then we have $\sqrt{P_c} = \sqrt{P_t}\cos(\phi_r) \approx 0.33$ and $\sqrt{P_r} = \sqrt{P_t}\sin(\phi_r) \approx 0.94$, and so (39) becomes

$$\tilde{r}(t) \approx 0.33 + j0.94\phi'_{\text{PN}}(t). \tag{42}$$

This is shown in the solid lines in the upper part of Figure 6. If $\theta(t) = \pi/16$ and the other parameters are the same, (39) becomes

$$\begin{aligned}\tilde{r}(t) &= \sqrt{P_c}\cos\left(\frac{\pi}{16}\right) - \sqrt{P_r}\sin\left(\frac{\pi}{16}\right)\phi'_{\text{PN}}(t)\\ &\quad + j\left(\sqrt{P_c}\sin\left(\frac{\pi}{16}\right) + \sqrt{P_r}\cos\left(\frac{\pi}{16}\right)\phi'_{\text{PN}}(t)\right) &(43)\\ &\approx 0.33 - 0.18\phi'_{\text{PN}}(t) + j(0.07 + 0.92\phi'_{\text{PN}}(t)). &(44)\end{aligned}$$

This is shown in the dashed lines in the upper part of Figure 6.

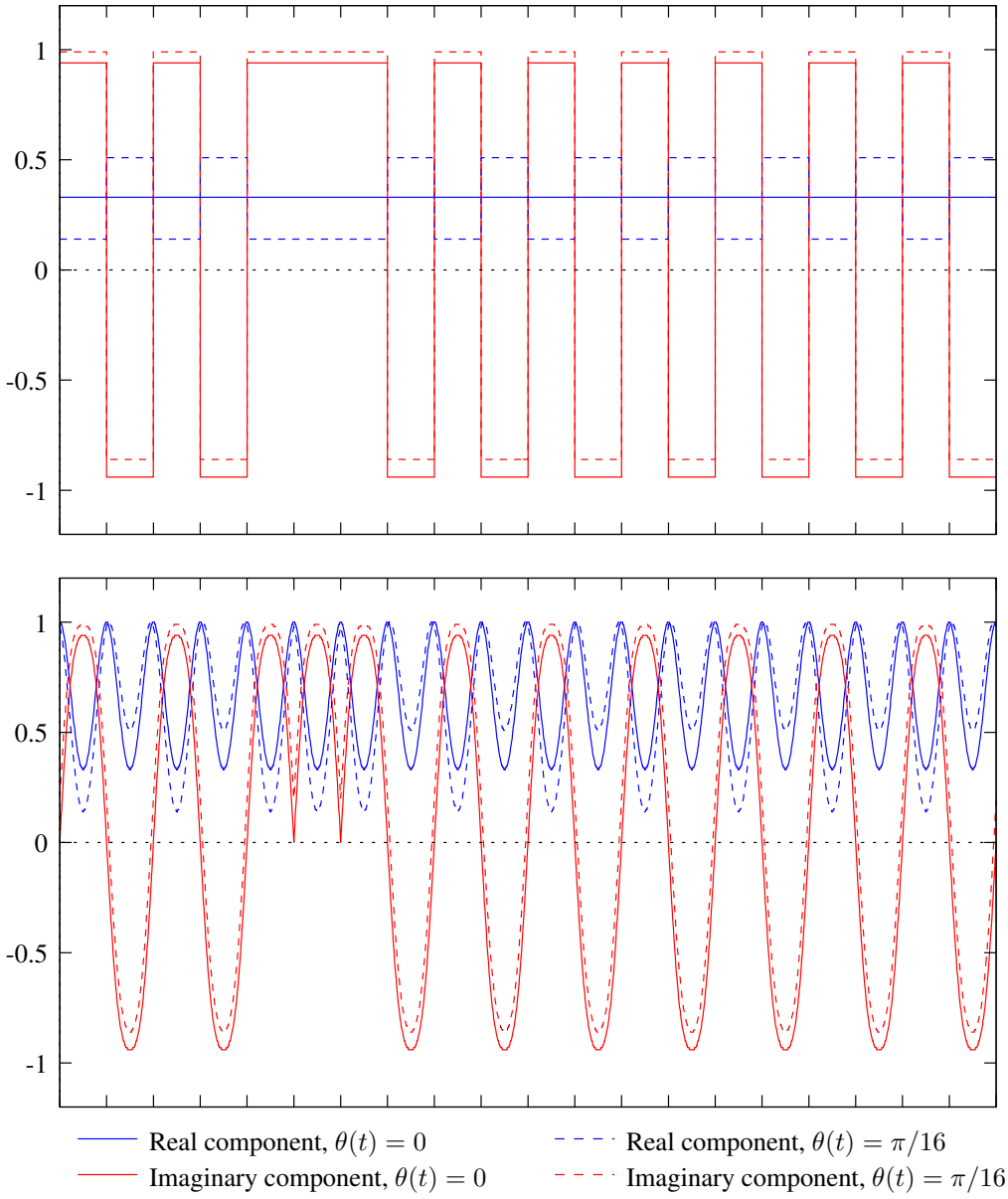


Figure 6. The complex baseband ranging signal of the T4B code used with the rectangular pulse shape (upper) and half-sine pulse shape (lower), $\phi_r = 1.23$, $P_t = 1$, $\theta(t) = 0$, **and** $N_0 = 0$.

On the other hand, when $p(t)$ is the half-sine pulse shape, the carrier and ranging signal powers in the fundamental sidebands are, respectively [2][1]

$$P_c = P_t \; J_0^2(\phi_r) \tag{45}$$

$$P_r = 2P_t \; J_1^2(\phi_r), \tag{46}$$

where $J_n(\cdot)$ is the nth order Bessel function of the first kind. In this case, the carrier and ranging signal powers P_c and P_r as defined here sum to less than the total transmitted power P_t. The higher-harmonic sidebands represent wasted power, because only the power in the fundamental sidebands contribute to the range measurements.

The lower part of Figure 6 illustrates $\tilde{r}(t)$ for the two examples above, except with $p(t)$ taking on the half-sine shape defined in (19).

B. Carrier Tracking of Residual Carrier Signal

The goal of carrier tracking is to lock onto and track $\theta(t)$ in (34). When the signal contains a residual carrier, the phase $\theta(t)$ is tracked with a phase-locked loop (PLL), as illustrated in Figure 7. The PLL takes in the complex-baseband received signal $\tilde{r}(t)$ from (34), mixes it with a complex carrier phase estimate $\tilde{w}(t) \triangleq \exp[-j\hat{\theta}(t)]$, takes the imaginary part, filters it, and sends the resulting error signal to a voltage controlled oscillator (VCO). Since this design is operating at baseband (DC), the VCO is not actually oscillating; instead, it is simply producing a complex conjugate phase estimate $\tilde{w}(t)$. The operation of the PLL is described by the pseudo-code in Algorithm 1.

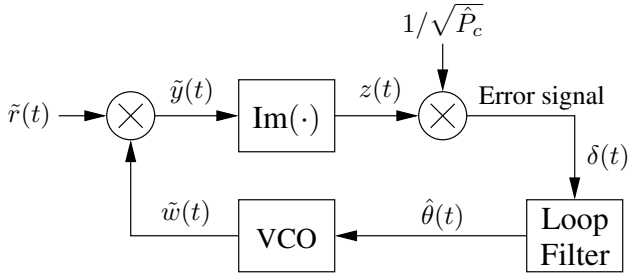


Figure 7. Continuous time model of phase-locked loop.

Using (34), the PLL in Figure 7 computes the phase-corrected output

$$\tilde{y}(t) \triangleq \tilde{r}(t)\tilde{w}(t) \tag{47}$$

$$= \sqrt{P_t}\exp[j(\phi_r\phi'_{\text{PN}}(t) + \Delta\theta(t))] + \tilde{n}'(t), \tag{48}$$

where

$$\Delta\theta(t) = \theta(t) - \hat{\theta}(t), \tag{49}$$

---

[1]In [2], the modulation index parameter is given in radians rms. Here, ϕ_r is the peak modulation index in radians. This accounts for a $\sqrt{2}$ difference in the formulas for carrier and signal powers.

Algorithm 1 Phase-locked loop.

1: **while** input present **do**
2: $\tilde{y}(t) \leftarrow \tilde{r}(t) \cdot \tilde{w}(t)$ {Mix input signal with VCO output}
3: $\delta(t) \leftarrow \mathrm{Im}\left(\tilde{y}(t)\right) / \sqrt{\hat{P}_c}$ {Compute error signal}
4: $\hat{\theta}(t) \leftarrow$ Loop filter$(\delta(t))$ {Apply loop filter to get carrier phase estimate}
5: $\tilde{w}(t) \leftarrow \mathrm{VCO}(\hat{\theta}(t))$ {Apply voltage δ to VCO}
6: **end while**

and where $\tilde{n}'(t) \triangleq \tilde{n}(t)\tilde{w}(t)$ has the same statistics as $\tilde{n}(t)$.

Thus,

$$z(t) \triangleq \mathrm{Im}(\tilde{y}(t)) \tag{50}$$

$$= \sqrt{P_t}\sin(\phi_r\phi'_{\mathrm{PN}}(t) + \Delta\theta(t)) + n'_s(t), \tag{51}$$

where $n'_s(t) = \mathrm{Im}(\tilde{n}'(t))$. If the PLL is operating with small phase error, $\Delta\theta(t) \ll 1$, then

$$\cos[\Delta\theta(t)] \approx 1 \tag{52}$$

$$\sin[\Delta\theta(t)] \approx \Delta\theta(t), \tag{53}$$

so that (51) becomes

$$z(t) = \sqrt{P_t}[\sin(\phi_r\phi'_{\mathrm{PN}}(t))\cos(\Delta\theta(t)) + \cos(\phi_r\phi'_{\mathrm{PN}}(t))\sin(\Delta\theta(t))] + n'_s(t) \tag{54}$$

$$\approx \sqrt{P_t}[\sin(\phi_r\phi'_{\mathrm{PN}}(t)) + \cos(\phi_r\phi'_{\mathrm{PN}}(t))\Delta\theta(t)] + n'_s(t) \tag{55}$$

$$= \sqrt{P_t}\cos(\phi_r\phi'_{\mathrm{PN}}(t))\Delta\theta(t) + n''_s(t), \tag{56}$$

where $n''_s(t) \triangleq \sqrt{P_t}\sin(\phi_r\phi'_{\mathrm{PN}}(t)) + n'_s(t)$. When $p(t)$ is the rectangular pulse shape, we have

$$z(t) \approx \sqrt{P_t}\cos(\phi_r)\Delta\theta(t) + n''_s(t) \tag{57}$$

$$= \sqrt{P_c}\Delta\theta(t) + n''_s(t). \tag{58}$$

The signal component present in $n''_s(t)$ is mitigated by the low-pass loop filter, but does give rise to a floor, as discussed in Section VII. This means that the output of the loop filter is proportional to the phase error $\Delta\theta(t)$ plus noise, which drives the VCO to continuously track the received phase.

C. Chip Timing Tracking with PLL

The final output of the carrier-tracking PLL is the mixer output $\tilde{y}(t)$ in (48), which is the received complex baseband signal with the varying carrier phase $\theta(t)$ mostly removed, i.e., except for the tracking error $\Delta\theta(t)$. The next step, as depicted in Figure 4, is to track the PN chip timing of $\phi'_{\mathrm{PN}}(t)$.

The offset of the received range code waveform $\phi'_{\mathrm{PN}}(t)$ relative to the spacecraft's reference waveform $\phi_{\mathrm{PN}}(t)$ can be decomposed as $2T(s+\epsilon)$ where s is an integer number of

range clock periods $2T$ (the range clock period is two chips), and $\epsilon \in [0, 1)$ is a fraction of a range clock period. That is,

$$\phi'_{\text{PN}}(t) = \phi_{\text{PN}}(t - 2T(s + \epsilon)). \tag{59}$$

Tracking the chip timing means tracking ϵ, based on observing the carrier phase-corrected signal $\tilde{y}(t)$ as given by (48).

When $p(t)$ is the half-sine shape, $\phi_{\text{PN}}(t)$ is highly correlated with $\phi_{\text{RC}}(t)$, as explained in Section II-A.2, and so for chip-tracking purposes, we may write

$$\begin{aligned}
\phi'_{\text{PN}}(t) &= \phi_{\text{PN}}(t - 2T(s + \epsilon)) && (60)\\
&\approx \phi_{\text{RC}}(t - 2T(s + \epsilon)) && (61)\\
&= \sin(2\pi f_{\text{RC}}(t - 2T(s + \epsilon))) && (62)\\
&= \sin(2\pi f_{\text{RC}}(t - 2T\epsilon)) && (63)\\
&= \sin(2\pi f_{\text{RC}} t + \theta_{\text{RC}}), && (64)
\end{aligned}$$

where (62) follows from (20), (63) follows because $2\pi f_{\text{RC}} 2Ts = 2\pi s$ is an even multiple of π, and in (64) we define $\theta_{\text{RC}} \triangleq -4\pi f_{\text{RC}} T\epsilon$, which is a phase we will track. Similarly, when $p(t)$ is the rectangular shape, we have

$$\begin{aligned}
\phi'_{\text{PN}}(t) \approx \phi_{\text{RC}}(t - 2T(s + \epsilon)) &= \text{sgn}(\sin(2\pi f_{\text{RC}} t + \theta_{\text{RC}})) && (65)\\
&= \frac{4}{\pi} \sum_{i=1,3,5}^{\infty} \frac{1}{i} \sin(2\pi i f_{\text{RC}} t + i\theta_{\text{RC}}), && (66)
\end{aligned}$$

where (66) is the Fourier series expansion of the square wave.

We now describe how $\tilde{y}(t)$ may be pre-processed into a signal from which θ_{RC} may be estimated using a PLL of the same type as described in the previous section. This pre-processing is shown in Figure 8.

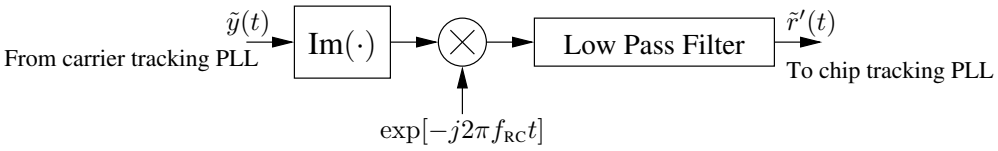


Figure 8. The carrier tracking loop output is down-converted.

When the carrier tracking loop is in lock, the signal component is contained almost entirely in the imaginary component. To simplify the analysis in the remainder of this subsection, we consider only the rectangular pulse shape. A locked carrier tracking loop means that $\Delta\theta(t) \ll 1$ and thus $\cos(\Delta\theta(t)) \approx 1$ and $\sin(\Delta\theta(t)) \approx \Delta\theta(t)$, so that (55) simplifies to

$$\text{Im}[\tilde{y}(t)] = \sqrt{P_r}\phi'_{\text{PN}}(t) + \sqrt{P_c}\Delta\theta(t) + n'_s(t). \tag{67}$$

This is mixed with a tone at the range clock frequency and low pass filtered. Because the transmitted range clock is coherent with the uplink frequency [2], the spacecraft can

use its recovered carrier frequency $\widehat{f}_c'$ to provide a Doppler rate-aided estimate of f_{RC}. This results in a PN chip tracking PLL input of

$$\begin{aligned}
\tilde{r}'(t) &\triangleq \text{LPF}\left[\text{Im}(\tilde{y}(t)) \cdot \exp[-j2\pi f_{\text{RC}} t]\right] && (68)\\
&= \text{LPF}\left[\left(\sqrt{P_r}\phi'_{\text{PN}}(t) + \sqrt{P_c}\Delta\theta(t) + n'_s(t)\right) \cdot (\cos(2\pi f_{\text{RC}} t) - j\sin(2\pi f_{\text{RC}} t))\right] && (69)\\
&= \text{LPF}\left[\sqrt{P_r}\phi'_{\text{PN}}(t)\cos(2\pi f_{\text{RC}} t) - j\sqrt{P_r}\phi'_{\text{PN}}(t)\sin(2\pi f_{\text{RC}} t)\right]\\
&\quad + \sqrt{P_c}\Delta\theta(t)\cos(2\pi f_{\text{RC}} t) - j\sqrt{P_c}\Delta\theta(t)\sin(2\pi f_{\text{RC}} t) + \tilde{n}'(t) && (70)\\
&\approx \sqrt{P_r}\phi'_{\text{PN}}(t)\cos(2\pi f_{\text{RC}} t) - j\sqrt{P_r}\phi'_{\text{PN}}(t)\sin(2\pi f_{\text{RC}} t) + \tilde{n}'(t) && (71)\\
&\approx \frac{2\sqrt{P_r}}{\pi}\sin(\theta_{\text{RC}}) - j\frac{2\sqrt{P_r}}{\pi}\cos(\theta_{\text{RC}}) + \tilde{n}'(t) && (72)\\
&= \frac{2\sqrt{P_r}}{\pi}\exp[j(\theta_{\text{RC}} - \pi/2)] + \tilde{n}'(t), && (73)
\end{aligned}$$

where in (71) we dropped the $\sqrt{P_c}$ terms because they are at a frequency near f_{RC} which will be removed by the low pass filter of the pre-processing step shown in Figure 8, where (72) follows from (66) and again the higher frequency terms ($i > 1$) are removed, and where $\tilde{n}'(t)$ is a complex noise term with two-sided power spectral density $N_0/4$ in each component.

The PN chip tracking PLL tracks θ_{RC} from (73) in the same way that the carrier tracking PLL tracks $\theta(t)$ from (39). The output of the PLL is an estimate $\hat{\theta}_{\text{RC}}$; the corresponding estimate $\hat{\epsilon}$ of the fractional chip delay is obtained from:

$$\hat{\theta}_{\text{RC}} = -4\pi\hat{f}_{\text{RC}} T\epsilon. \tag{74}$$

At this point, soft symbols of the chips may be computed by

$$v_k \triangleq \int_{(k-1)T+\hat{\epsilon}}^{kT+\hat{\epsilon}} \text{Im}\left(\tilde{y}(t)\right)\, dt, \tag{75}$$

where $\tilde{y}(t)$ is the original carrier-phase-adjusted input to the chip tracking loop. Thus, the final output of the chip timing tracking loop is a set of discrete-time soft symbols v_k.

D. PN Sequence Acquisition

The sequence of match filtered chips $\{v_k\}$ in (75) is a noisy delayed version of the PN sequence $\{d_k\}$:

$$\begin{aligned}
v_k &= d_{k-2s} + n_k && (76)\\
&= d_{k-u} + n_k, && (77)
\end{aligned}$$

where n_k is additive white Gaussian noise, s is an unknown integer offset of range clock periods from (59), and $u \triangleq 2s$ is an integer offset number of chip periods. The offset u is an even integer because the chip tracking timing has resolved the fractional timing with respect to the range clock period, which is two chips. Nevertheless, in the following, we shall not make use of the even-ness of u.

The component structure of the PN sequence enables acquisition of the range code without correlating against its full million-long period. Instead, only correlators for each of the shorter *component* codes are needed. A functional block diagram of a correlator for the jth component code is shown in Figure 9. The match filtered symbols v_k are correlated with each shifted version of the jth component code. Each shifted correlation is accumulated (summed) and the index u_j of the largest accumulator is chosen:

$$u_j = \underset{u}{\operatorname{argmax}} \sum_k v_{k-u} C'_j(k). \tag{78}$$

The sum uses sufficiently many terms to achieve the desired probability of correct acquisition. The index u_j gives the offset, modulo L_j, between the overall range code and the locally generated PN sequence. This is done for each sequence, resulting in the indices u_1, u_2, u_3, u_4, u_5, and u_6.

Once component codes are acquired (i.e., u_1, u_2, u_3, u_4, u_5, and u_6 are determined), the overall delay may be calculated by applying the Chinese Remainder Theorem [8], as follows. Let

$$m_j \triangleq L/L_j \tag{79}$$

$$m_j^{-1} \mod L_j \triangleq \{n \in \{0, 1, \ldots, L_j - 1\} : n m_j = 1 \mod L_j\} \tag{80}$$

$$a_j \triangleq m_j (m_j^{-1} \mod L_j). \tag{81}$$

For each $j \in \{1, 2, 3, 4, 5, 6\}$, the values of L_j, m_j, $m_j^{-1} \mod L_j$, and a_j are listed in Table 2. By the Chinese Remainder Theorem, the estimate $\hat{u}$ of the delay u in the length-L range code is given by

$$\hat{u} = \sum_{j=1}^{6} a_j u_j \mod L. \tag{82}$$

The offset $\hat{u}$ indicates the number of chips (symbols) by which the received and locally generated sequence differ, and the estimate will be correct whenever the component code acquisitions are correct, in which case $\hat{u}$ will be even as described above. The estimate in units of range clock periods is simply $\hat{s} = \hat{u}/2$.

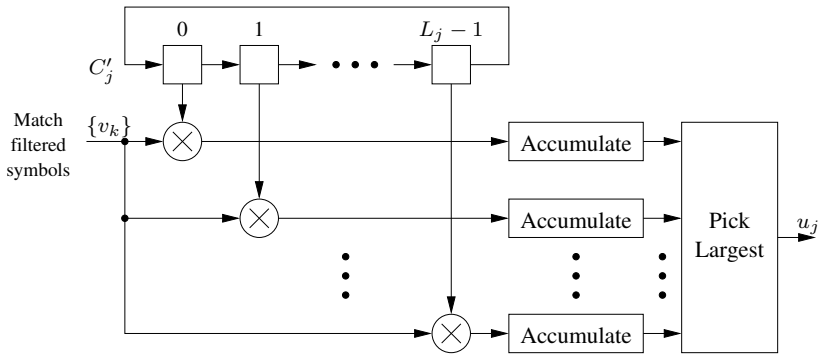


Figure 9. Correlator for the jth component code. This structure is repeated for $j = 1, 2, 3, 4, 5, 6$.

Table 2. Constants used in PN range code acquisition.

j	L_j	m_j	$(m_j^{-1} \bmod L_j)$	a_j
1	2	504,735	1	504,735
2	7	144,210	5	721,050
3	11	91,770	7	642,390
4	15	67,298	2	134,596
5	19	53,130	16	850,080
6	23	43,890	4	175,560

As an example of this calculation, suppose the input to the receiver is a PN range code with offset $u = 123,456$ chips, and the receiver is to determine the value of u. If each component code is properly acquired using a correlator as shown in Figure 9, then for each j the receiver would select u_j satisfying $u_j = u \mod L_j$, or $u_1 = 0$, $u_2 = 4$, $u_3 = 3$, $u_4 = 6$, $u_5 = 13$, and $u_6 = 15$. Applying (82), we have

$$\begin{aligned}\hat{u} &= 504,735 \times 0 + 721,050 \times 4 + 642,390 \times 3 + 134,596 \times 6 + 850,080 \times 13 \\ &\quad + 175560 \times 15 \mod 1,009,470 \\ &= 123,456,\end{aligned}$$

which equals u as expected.

E. Range Code Phase Computation

Using the signal processing described in the preceding sections, the spacecraft has an estimate $\hat{s}$ and $\hat{\epsilon}$ for any time t of the spacecraft's time reference. From (59), its estimate of the timing offset between the received range code waveform $\phi'_{\text{PN}}(t)$ and the spacecraft reference PN waveform $\phi_{\text{PN}}(t)$ is $2T(\hat{s} + \hat{\epsilon})$. The range code phase $\psi_s(t)$, in radians, is related to the time offset estimate by

$$\frac{\psi_s(t)}{2\pi} = \frac{2T(\hat{s} + \hat{\epsilon})}{LT}. \tag{83}$$

That is, one full cycle of the phase (2π radians) corresponds to a time offset of one full period (L chips of duration T) of the range code waveform. Since the right-hand side is in $[0, 1)$, the spacecraft is computing a wrapped phase, i.e., $\psi_s(t) \in [0, 2\pi)$.

As described in in Section II of [1], it is more convenient to represent the wrapped phase $\psi_s(t)$ in units of chips, instead of radians, which results in the simpler expression

$$\psi_S(t) = 2(\hat{s} + \hat{\epsilon}), \tag{84}$$

where, again, it should be understood that this is a wrapped phase, not the unwrapped phase needed in (2). The ground processing at the DSS is responsible for unwrapping the phases measured at the spacecraft, by looking at the entire sequence of wrapped phases transmitted to the ground and using prior knowledge of the spacecraft's range.

On board the spacecraft, the ground reference time is not known, and it may be that no stable time reference is available at all. The spacecraft does not need to know time; it merely needs to be able to compute and store the (wrapped) value of $\psi_S(t)$ from (84) at the time that a downlink telemetry codeword begins being transmitted. (That time, in the *ground's* reference time frame, occurs at time t_S, but the spacecraft need not know this.) Later, this value of $\psi_S(t_S)$ is associated as in (2) with another phase $\psi_T(t_R)$, measured at the DSS at the time t_R when the same telemetry codeword arrives at the receiver. Then this process is repeated for additional sample times t_S at the start of other downlink telemetry codewords, to be associated with their respective ground reception times t_R.

IV. Signal Processing in the Downlink Receiver

The ground station receives a noisy, delayed version of (21), which can be represented as

$$r(t) = s'(t) + n(t). \tag{85}$$

where $s'(t)$ is a delayed version of the transmitted signal $s(t)$, and $n(t)$ is a white Gaussian noise process with two-sided spectral density $N_0/2$. The argument t in (85) is the local ground time reference. As we saw for the uplink signal, the time-varying downlink delay introduces a Doppler-shift in the received carrier frequency, as well as a delay in the time of the arriving telemetry.

The ground receiver down-converts, digitizes, and tracks the downlink signal. A functional block diagram of the ground receiver is shown in Figure 10. Again, for now we ignore the digitization and describe the signal processing using a continuous-time notation in this section. Section V will describe the discrete-time signal processing. In the following subsections, we describe each box shown in the figure.

A. Down-Conversion to Baseband

The telemetry signal is down-converted using the same structure as in Figure 5, except that the local oscillator has frequency $\widehat{f}'_c$, accounting for the Doppler seen on the downlink. When the pulse shape is rectangular, as is common for the downlink signaling, this results in the complex baseband signal

$$\tilde{r}(t) = \sqrt{P_c}e^{j\theta(t)} + j\sqrt{P_d}e^{j\theta(t)}\phi'_{\rm TM}(t) + \tilde{n}(t), \tag{86}$$

where $P_c = P_t\cos^2(\phi_d)$, $P_d = P_t\sin^2(\phi_d)$, and $\phi'_{\rm TM}(t)$ is a delayed version of $\phi_{\rm TM}(t)$. This is the same form as we saw for the uplink, given by (39), except that here we have a data modulation index of ϕ_d instead of the ranging modulation index ϕ_r, and we have a downlink modulating signal $\phi'_{\rm TM}(t)$ arriving at $\mathcal{P}_{\rm R}$ instead of an uplink modulating signal $\phi_{\rm PN}(t)$ arriving at $\mathcal{P}_{\rm S}$.

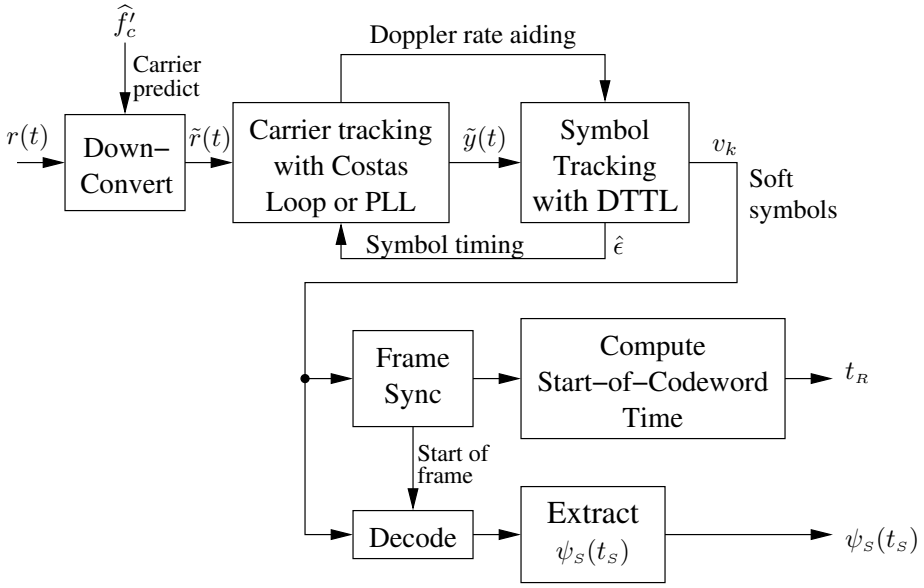


Figure 10. The downlink receiver on the ground.

B. Carrier Tracking with PLL or Costas Loop

If a residual carrier is present, the ground receiver may use a PLL to track the carrier phase, exactly as described for the uplink receiver in Section III-B, resulting in the output $\tilde{y}(t)$. On the other hand, when $\phi_d \to \pi/2$, we have $P_c \to 0$, and the PLL filter input (58) contains only noise. Therefore, a different approach is needed to track the phase of a suppressed carrier signal.

A well-known technique for tracking the phase of a suppressed carrier signal is the Costas loop, which employs both the imaginary and real components of the phase-corrected downconverted signal $\tilde{y}(t)$ to generate the error signal. A block diagram of the Costas loop for continuous time signals is shown in Figure 11. We assume a rectangular pulse shape is used, as is typical of downlink transmissions. When the carrier is suppressed,

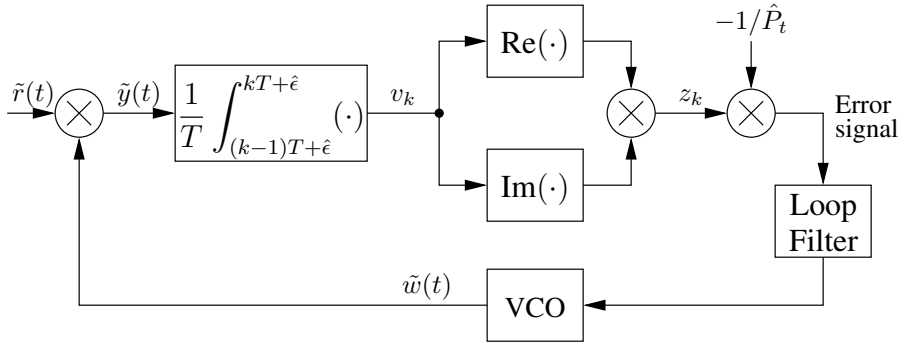


Figure 11. Costas loop for a continuous-time, complex input.

we have $\phi_d = \pi/2$, and so $P_c = 0$ and $P_d = P_t$ in (86), so that when $\tilde{r}(t)$ is multiplied by $\tilde{w}(t) = e^{-j\hat{\theta}(t)}$ we have

$$\tilde{y}(t) = j\sqrt{P_t}e^{j\Delta\theta(t)}\phi'_{\text{TM}}(t) + \tilde{n}'(t). \tag{87}$$

Assuming P_t and $\Delta\theta(t)$ are relatively constant over the duration of a symbol, and further, that the symbol timing is known to a good accuracy, we may integrate over the interval of a symbol to obtain

$$v_k \triangleq \frac{1}{T}\int_{(k-1)T+\hat{\epsilon}}^{kT+\hat{\epsilon}} \tilde{y}(t)\,dt \tag{88}$$

$$= j\sqrt{P_t}e^{j\Delta\theta_k}d_k + \tilde{n}'_k, \tag{89}$$

where $\Delta\theta(t) = \Delta\theta_k$ within the kth symbol interval, and where

$$\tilde{n}'_k \triangleq \frac{1}{T}\int_{(k-1)T+\hat{\epsilon}}^{kT+\hat{\epsilon}} \tilde{n}'(t)\,dt \triangleq n'_{c,k} + jn'_{s,k} \tag{90}$$

is the noise integrated over a symbol interval. The real and imaginary parts of v_k are

$$\text{Re}[\tilde{v}_k] = -\sqrt{P_t}\sin(\Delta\theta_k)d_k + n'_{c,k} \tag{91}$$

$$\text{Im}[\tilde{v}_k] = \sqrt{P_t}\cos(\Delta\theta_k)d_k + n'_{s,k}, \tag{92}$$

which are multiplied to form

$$z_k \triangleq \text{Re}[\tilde{v}_k]\cdot\text{Im}[\tilde{v}_k] \tag{93}$$

$$= -\frac{P_t}{2}\sin(2\Delta\theta_k)d_k^2 + n''_k \tag{94}$$

$$\approx -P_t\Delta\theta_k + n''_k, \tag{95}$$

where in (94) we used $2\sin(x)\cos(x) = \sin(2x)$ and where we let the cross terms and squared noise term be denoted by

$$n''_k \triangleq \sqrt{P_t}\cos(\Delta\theta_k)d_k n'_{c,k} - \sqrt{P_t}\sin(\Delta\theta_k)d_k n'_{s,k} + n'_{c,k}n'_{s,k}. \tag{96}$$

In (95) we used $d_k^2 = 1$ and assumed $\Delta\theta_k \ll 1$, so that $\sin(2\Delta\theta_k) \approx 2\Delta\theta_k$. The $-P_t$ is a scale factor that can be removed, and as a result, the signal into the loop filter is proportional to the phase error, plus higher frequency terms that are attenuated by the loop filter, and noise.

The final output of the Costas loop is the mixer output $\tilde{y}(t)$ in (87), which is the received complex baseband signal with the varying carrier phase $\theta(t)$ removed.

C. Symbol Timing Synchronization with DTTL

The next stage in the ground receiver is to recover the symbol timing. The received downlink waveform includes an unknown symbol timing offset ϵT, where $\epsilon \in [0, 1)$ is a

fraction of a symbol duration T. Tracking the symbol timing means tracking ϵ, based on observing the carrier phase-corrected signal given by

$$\tilde{y}(t) = \begin{cases} \sqrt{P_c}e^{j\Delta\theta(t)} + j\sqrt{P_d}e^{j\Delta\theta(t)}\phi'_{\text{TM}}(t) + \tilde{n}'(t), & \text{Carrier tracking with PLL} \\ j\sqrt{P_d}e^{j\Delta\theta(t)}\phi'_{\text{TM}}(t) + \tilde{n}'(t), & \text{Carrier tracking with Costas loop.} \end{cases} \tag{97}$$

As noted in Section III-C for the uplink signal, after carrier tracking most of the signal is in the imaginary part of $\tilde{y}(t)$, and analogous to (67), we have

$$\text{Im}[\tilde{y}(t)] = \sqrt{P_d}\phi'_{\text{TM}}(t) + \sqrt{P_c}\Delta\theta(t) + n'_s(t). \tag{98}$$

If we further assume $P_c = 0$ or $\Delta\theta(t) \approx 0$, this simplifies to

$$\text{Im}[\tilde{y}(t)] = \sqrt{P_d}\phi'_{\text{TM}}(t) + n'_s(t). \tag{99}$$

The symbols are tracked with a data-transition tracking loop (DTTL). A block diagram of the DTTL is shown in Figure 12. At the input is $\text{Im}[\tilde{y}(t)]$, as given in (99). The upper arm forms the sequence $\{v_k\}$ given by

$$v_k \triangleq \int_{(k-1)T+\hat{\epsilon}}^{kT+\hat{\epsilon}} \text{Im}[\tilde{y}(t)]\, dt. \tag{100}$$

This is an in-phase integration resulting in the soft symbol estimate v_k. Hard decisions are formed by $\hat{d}_k \triangleq \text{sgn}(v_k)$, followed by a transition detection

$$D_k \triangleq \frac{\hat{d}_{k-1} - \hat{d}_k}{2}. \tag{101}$$

In this way, D_k is 1 for a positive-to-negative transition, 0 for no transition, and -1 for a negative-to-positive transition.

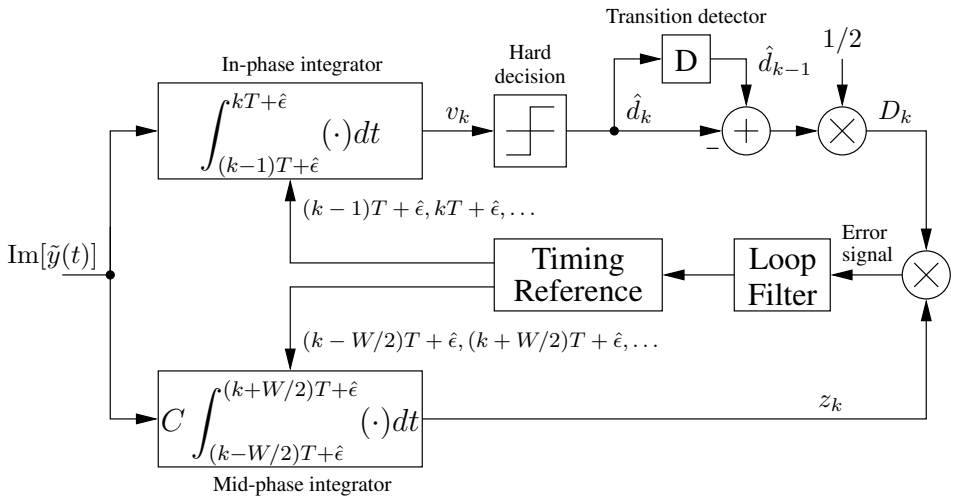


Figure 12. The DTTL.

The lower arm forms the sequence $\{z_k\}$ given by

$$z_k \triangleq C \int_{(k-W/2)T+\hat{\epsilon}}^{(k+W/2)T+\hat{\epsilon}} \mathrm{Im}[\tilde{y}(t)]\, dt. \tag{102}$$

where $W = 2^{-n}$ is a window fraction with $n \in \{0, 1, 2, 3, 4\}$, and where $C = 1/(2\sqrt{P_d})$ is a normalization factor. This is a mid-phase integration resulting in an estimate of a timing offset.

The sign-corrected error signal is formed by the product of the upper and lower arms, $D_k z_k$, which is the input to the loop filter that averages the instantaneous error signal to estimate the delay and produces a filtered input to the timing reference, whose timing signal is adjusted to shift the in-phase and mid-phase interval boundaries, reducing the relative delay between the timing reference clock and the received symbols.

The operation of the DTTL can be explained with the help of Figure 13(a), which shows a section of the symbol-sequence consisting of rectangular pulses of known duration T, where for discussion purposes the additive noise is assumed to be negligible. This figure illustrates a case where the timing reference outputs are synchronized with the received signal.

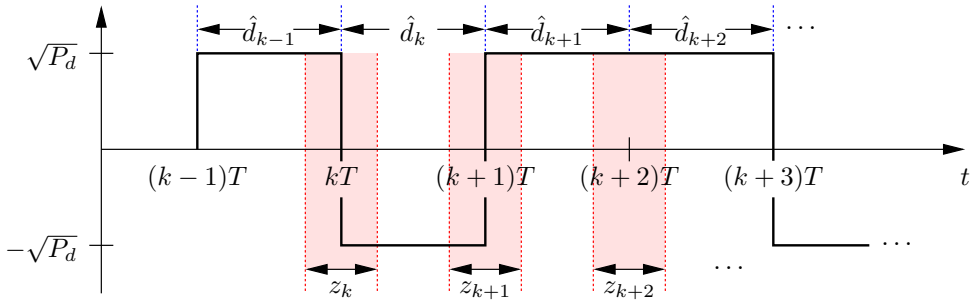
In Figure 13(a), the symbol decisions $\hat{d}_{k-1}, \hat{d}_k, \hat{d}_{k+1}, \ldots$ are made with in-phase, full-symbol-duration integrations over the intervals indicated by blue dashed lines. The accompanying table shows the symbol decisions for the example considered. Likewise, the lower arms computes $z_{k-1}, z_k, z_{k+1}, \ldots$. In this example the window fraction is $W = 1/2$, which means the mid-phase integration occurs over a duration of $T/2$. In this example the $z_k = 0$ zero whenever there is a transition because the loop is synchronized. Thus, the error signal $D_k z_k = 0$ for all k.

Figure 13(b) illustrates what happens when the loop's time reference is slow by δ. Provided δ is small, the symbol decisions remain correct, as indicated in the table. Because of the timing offset, the mid-phase integral in the lower arm will be $\pm\delta/2$ when a digital transition is present, depending on the direction of the transition. Thus, the error signal $D_k z_k$ is δ whenever a digital transition is present, and zero otherwise, as indicated in the table. We note that in the noise-free case we have described the error signal does not depend on W. When noise is present, the variance of z_k is proportional to W, and thus, it is advantageous to use a lower value of W provided the mid-phase integral still captures the region of the digital transition.

The final output of the DTTL is the soft symbol v_k at the output of the in-phase integrator in the upper arm.

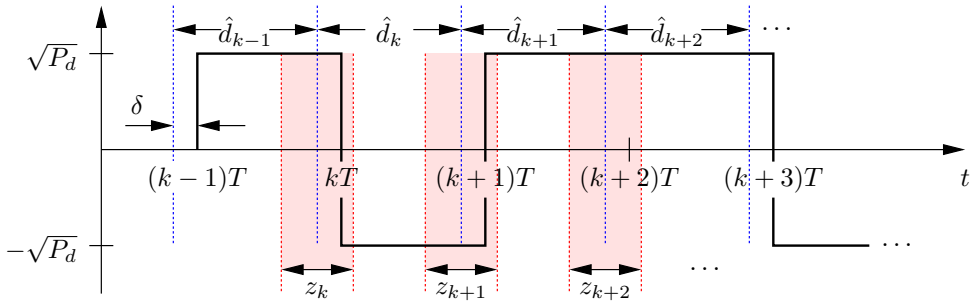
D. Determining the Parameters Used in the Time Delay Computation

The output of the DTTL is a sequence of soft symbols $\{v_k\}$. The first symbol of a codeword must be identified in order for proper decoding to take place. This is



Index	$\hat{d}_{\text{Index}}$	D_{Index}	z_{Index}	Error signal
$k-1$	1	-	-	-
k	-1	1	0	0
$k+1$	1	-1	0	0
$k+2$	1	0	$WT/2$	0

(a) When operating in lock.



Index	$\hat{d}_{\text{Index}}$	D_{Index}	z_{Index}	Error signal
$k-1$	1	-	-	-
k	-1	1	δ	δ
$k+1$	1	-1	$-\delta$	δ
$k+2$	1	0	$WT/2$	0

(b) When operating with a timing offset of δ.

Figure 13. The in-phase and mid-phase integration intervals used by the DTTL.

accomplished by performing a running correlation, or related statistic, with a known synchronization marker that is inserted between each codeword [9].

When the frame synchronization marker is found, the receiver records a computed time-tag t_R associated with when the first symbol of the codeword arrived at $\mathcal{P}_{\mathrm{R}}$. This can be done extremely accurately by using the frame synchronizer's determination of the integer symbol offset and the DTTL's estimate of the fractional delay ϵ, in relation to the 1-pps or other accurate timing reference. For example, in open loop recorded data, an accurate time-stamp may occur once per second, and any sample (or even a time between samples) may be associated with a time which is a linear interpolation of 1-pps epochs occurring before and after the sample. This type of interpolation is generally accurate to less than a nanosecond.

Identifying the frame synchronization marker also enables the decoder to operate on the codeword immediately following it. Once decoded, any stored spacecraft range code phase measurement $\psi_S(t_S)$ within the data is extracted.

This results in the final outputs of the downlink receiver, t_R and $\psi_S(t_S)$, shown in Figure 10. The transmitted phase $\psi_T(t_R)$ and phase derivative $\left\{\dot{\psi}_T(t), t \in [t_T, t_R]\right\}$ are retrieved from DSS records. At this point we will have collected all the measurements needed to complete the time delay computation in (2). The final step of converting/-calibrating this time delay $(t_R - t_T)$ to the two-way geometric range is described in the previous article [1].

V. Digital Processing Implemented in Ranging Software

As mentioned at the beginnings of Sections III and IV, the continuous-time signal processing block diagrams, depicted in Figures 4 and 10 for the uplink and downlink, respectively, are simplifications in that they omit the analog-to-digital (A-to-D) conversions that take place in each processing chain. After downconversion, the complex analog waveforms are sampled via short-term integration at a rate commensurate with the Nyquist criterion and specific design constraints:

$$\tilde{r}_i \triangleq \frac{1}{T_S} \int_{(i-1)T_S}^{iT_S} \tilde{r}(t)\, dt. \tag{103}$$

The actual tracking loops past that point are therefore digital versions of the tracking loops described in Sections III and IV. Figures 14 and 15 are block diagrams for the digitized versions of the uplink and downlink receivers.

We implemented MATLAB-based software to simulate the operation and performance of a telemetry ranging system using digitized tracking loops. The remainder of this section gives equations and pseudo-code specifying the operation of the major software functions depicted in Figures 14 and 15 past the point of A-to-D conversion. Then, in Section VI we describe two useful software tools built from these functional blocks. Finally, in Section VII we provide performance analyses, comparing software-simulated

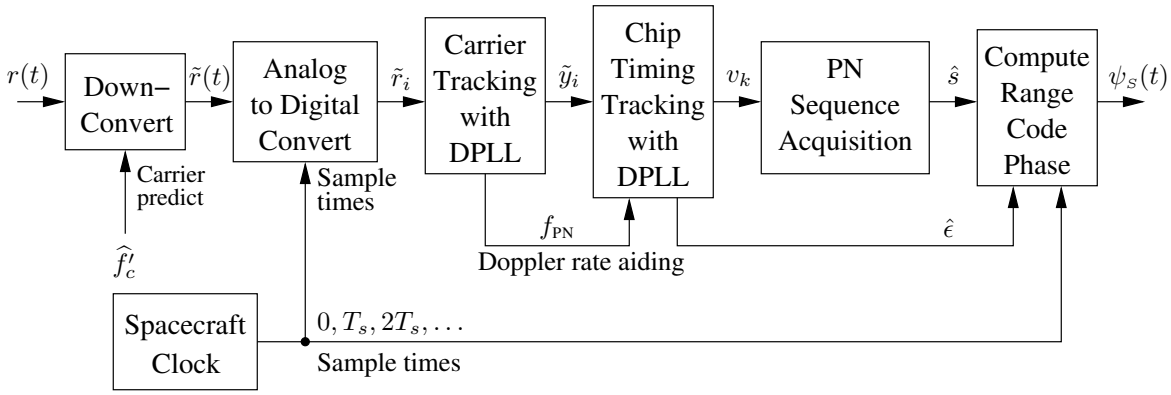


Figure 14. The digitized uplink receiver.

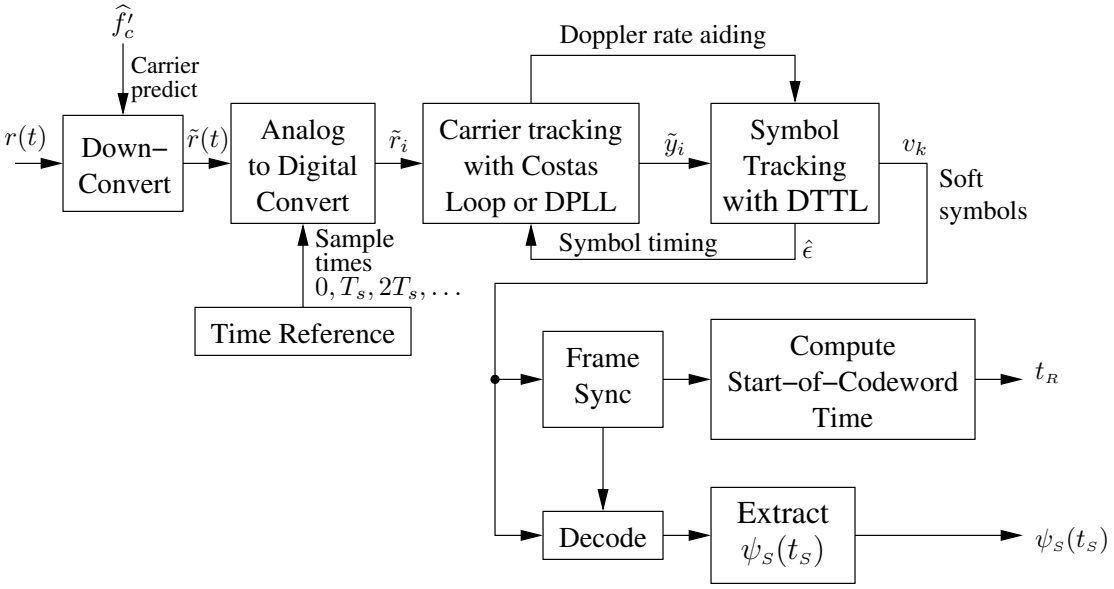


Figure 15. The digitized downlink receiver.

performance of the discrete-time tracking loops with Cramér-Rao bounds derived from the corresponding continuous-time models.

A. Residual Carrier Tracking Using a DPLL

Figure 16 illustrates a digital phase-locked loop (DPLL) for tracking the phase of a residual carrier from a discrete-time, complex baseband signal. The software's DPLL function is used by both the uplink and downlink receivers unless the downlink telemetry signal is suppressed-carrier.

The digital loop structure is obtained from the analog model by discretizing the analog signals, which are then processed digitally by the DPLL. Using the sample times $t_i \triangleq iT_S$, the discrete-time input to the loop is given by:

$$\tilde{r}_i = \sqrt{P_t}\exp[j(\phi_r\phi_{\text{PN}}(t_i) + \theta_i)] + \tilde{n}_i, \tag{104}$$

where $\theta_i \triangleq \theta(t_i)$.

The discrete representation of the carrier tracking loop model is shown in Figure 16. As in the analog PLL, the input signal to the DPLL is first counter-rotated with the current estimate of the carrier phase, in this case sample by sample, by $\tilde{w}_k$. The $\tilde{w}_k$ term is updated by the numerically-controlled oscillator (NCO) at the loop update rate, R_{update}, which is slower than the sample rate $R_{\text{samp}} = 1/T_S$ by a factor $K \triangleq R_{\text{samp}}/R_{\text{update}}$. The counter-rotated complex signal is then accumulated and averaged over K samples, i.e., an interval of length $T_u \triangleq KT_S$. Assuming a rectangular pulse shape, the imaginary component of this summation is

$$z_k = \sqrt{P_c}\Delta\theta_k + n''_{s,k}, \tag{105}$$

which is the discrete-time version of (58). Normalizing this by $1/\sqrt{\hat{P}_c}$ allows one to extract the residual phase (or phase error) for the current update interval k

$$\Delta\theta_k = \frac{1}{\sqrt{\hat{P}_c}} \cdot z_k. \tag{106}$$

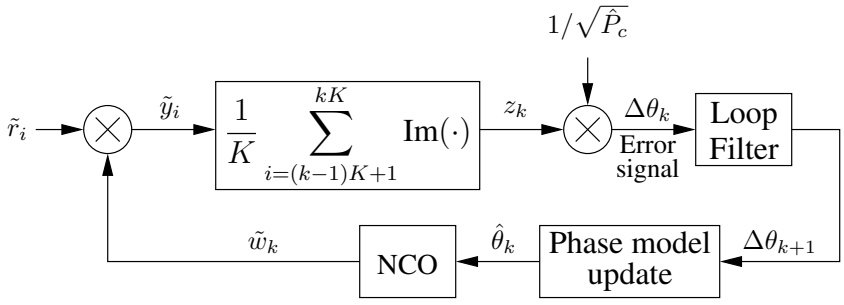


Figure 16. DPLL.

In our simulation, the model phase for the $(k+1)^{th}$ interval is estimated from the current phase estimate $\hat{\theta}_k$ and residual phase $\Delta\theta_k$ using a conventional second order digital loop filter, as described in [10].

$$\hat{\theta}_{k+1} = \hat{\theta}_k + K_1 \Delta\theta_k + K_2 \sum_{n=1}^{k} \Delta\theta_n \tag{107}$$

The coefficients of the filter are calculated based on the loop filter bandwidth B_L and update interval length T_u (see [10]: $K_1 = \frac{8}{3} B_L T_u$, $K_2 = \frac{1}{2} K_1^2$. These gain coefficients correspond to the continuous update, standard underdamped model in [10], which is accurate when $B_L T_u < 0.02$. Hence we shall use the continuous time model to analyze the phase locked loops in Section VII.

The operation of the DPLL in Figure 16 is described by the following pseudo-code. Note that the loop filter operates at a lower rate than the sampling rate, and therefore we use different indices (i and k, respectively) for the input signal and NCO output.

Algorithm 2 Digital Phase-locked loop (DPLL).

while input present **do**
  $\tilde{y}_i \leftarrow \tilde{r}_i \cdot \tilde{w}_k$ {Mix input signal with NCO output}
  $z_k \leftarrow \frac{1}{K} \sum_{i=(k-1)K+1}^{kK} \mathrm{Im}(\tilde{y}_i)$ {Take imaginary part and average over K samples}
  $\Delta\theta_k \leftarrow \frac{1}{\sqrt{\hat{P}_c}} \cdot z_k$ {Normalize to extract residual phase}
  $\hat{\theta}_{k+1} \leftarrow \hat{\theta}_k + K_1 \Delta\theta_k + K_2 \sum_{n=1}^{k} \Delta\theta_n$ {Estimate phase for next interval using loop filter}
  $\tilde{w}_{k+1} \leftarrow \exp[-j\hat{\theta}_{k+1}]$ {Apply model phase estimate to NCO for next interval}
end while

B. Suppressed Carrier Tracking with Costas Loop

The downlink signal modulation format can be configured by setting the modulation index from the GUI. The options allow either a fully suppressed carrier, i.e., $\phi_d = \pi/2$ radians, or a residual carrier, i.e., $\phi_d < \pi/2$ radians. A DPLL (as described in the previous subsection) is used when a residual carrier is present, and the software automatically selects a Costas loop for a fully suppressed carrier.

Figure 17 shows the block diagram of the digital implementation of the Costas loop. What was an integral of $\tilde{y}(t)$ in the continuous-time in Figure 11 has become an equivalent discrete-time summation. Here, N_s is the number of samples per symbol, $n_\epsilon \triangleq \lfloor \epsilon \cdot N_s \rfloor$ is the integer part of the sample-index of the timing offset ϵ, and $\delta_\epsilon \triangleq \epsilon \cdot N_s - n_\epsilon$ is the fractional sample of the timing offset. The weighted sum indicated in the figure enables the inclusion of only a fraction of the first and last sample of a symbol in the summation. After multiplying the real and imaginary parts, the average detected phase signal is then normalized to obtain the residual phase $\Delta\theta_k = -z_k/\hat{P}_t$. Note that the normalization factor is $\hat{P}_t$, and not $\sqrt{\hat{P}_t}$, because phase error is obtained

by multiplying the I and Q components of the signal. The model phase for the next update interval is estimated using a second order loop filter based on (107).

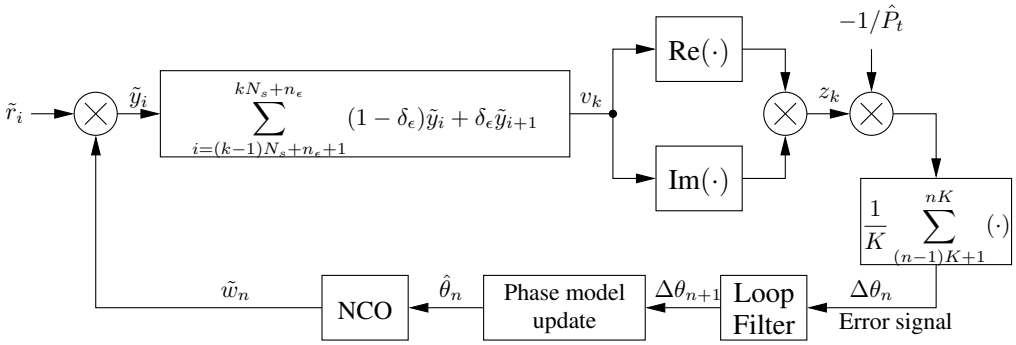


Figure 17. The discrete-time Costas loop, as implemented in the simulation.

The operation of the Costas loop in Figure 17 is described by the following pseudo-code. This pseudo-code does not include the optional additional summation prior to the loop

Algorithm 3 Costas loop

while input present **do**
  $\tilde{y}_i \leftarrow \tilde{r}_i \cdot \tilde{w}_k$ {Mix input signal with NCO output}
  $v_k = \sum_{i=(k-1)N_s+n_\epsilon+1}^{kN_s+n_\epsilon}(1-\delta_\epsilon)\cdot \mathrm{Im}(\tilde{y}_i) + \delta_\epsilon \cdot \mathrm{Im}(\tilde{y}_{i+1})$
  $z_k = \mathrm{Re}(\tilde{v}_k)\cdot \mathrm{Im}(\tilde{v}_k)$ {Average phase error for current interval}
  $\Delta\theta_k \leftarrow -\frac{1}{\hat{P}_t}\cdot z_k$ {Normalize to extract residual phase}
  $\hat{\theta}_{k+1} \leftarrow \hat{\theta}_k + K_1\Delta\theta_k + K_2\sum_{n=1}^{k}\Delta\theta_n$ {Next phase estimate, using loop filter}
  $\tilde{w}_{k+1} \leftarrow \exp(-j\hat{\theta}_{k+1})$ {Apply model phase estimate to NCO for next interval}
end while

filter. Such a summation, when used, enables the loop update rate to be slower than the symbol rate, if desired.

C. DTTL for Uplink Chip Timing Recovery or Downlink Symbol Timing Recovery

The fractional component of the ranging signal phase, ϵ, can be recovered using either a traditional DTTL or a DPLL loop. The user may select which loop type to use by toggling the loop type button in the spacecraft receiver control panel of the GUI. When DPLL is selected, the uplink signal is first pre-multiplied by a sinusoidal waveform of half the chip rate. The phase of the composite signal is proportional to the timing offset of the ranging signal and can be recovered using the standard DPLL loop described in Section V-A. We report the performance of chip tracking using DPLL in Section VII-E.1.

The DTTL can be used to recover the fractional timing for both uplink PN ranging signal and downlink telemetry signal. Figure 18 shows the block diagram of the digital

implementation of a DTTL. As a digital implementation of Figure 12, our software uses weighted sums to compute the in-phase and mid-phase integrations. Another difference worth noting is that the timing estimate is updated at a rate that is potentially lower than the symbol(chip) rate.

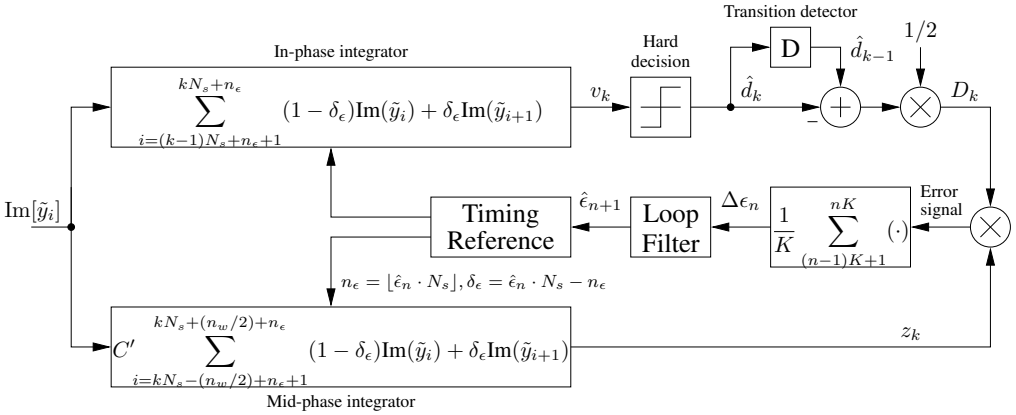


Figure 18. The DTTL, as implemented in our telemetry ranging software.

For a residual carrier, we extract the quadrature component of the demodulated signal as the input to the loop. When the carrier is fully suppressed, the in-phase component contains the data modulation and is therefore used as the input to the DTTL. Using the timing offset estimate $\hat{\epsilon}$ as reference, the in-phase and mid-phase integrators calculate the weighted sum of $\tilde{y}(i - n_\epsilon)$ and $\tilde{y}(i - n_\epsilon + 1)$ to obtain soft symbol estimate v_k and timing error estimate z_k. As before, we let let n_ϵ (samples) $= \lfloor \epsilon \cdot N_s \rfloor$ be the integer component, and $\delta_\epsilon = \epsilon \cdot N_s - n_\epsilon$ be the fractional component of the timing offset. The in-phase (soft symbol) and mid-phase integrations are, respectively,

$$v_k = \sum_{i=(k-1)N_s+n_\epsilon+1}^{kN_s+n_\epsilon} (1-\delta_\epsilon) \cdot \mathrm{Im}(\tilde{y}_i) + \delta_\epsilon \cdot \mathrm{Im}(\tilde{y}_{i+1}) \tag{108}$$

$$z_k = C' \sum_{i=kN_s-\frac{n_w}{2}+n_\epsilon+1}^{kN_s+\frac{n_w}{2}+n_\epsilon} (1-\delta_\epsilon) \cdot \mathrm{Im}(\tilde{y}_i) + \delta_\epsilon \cdot \mathrm{Im}(\tilde{y}_{i+1}). \tag{109}$$

Here N_s is the number of samples per symbol (chip), and $n_w = W \cdot N_s$ is the number of samples in the mid-phase integration window. Note that we use C' for the normalization factor in mid-phase integrator to differentiate from the continuous-time model, i.e., $C' = 1/(2n_w\sqrt{P_d})$.

Data transition D_k is detected from the estimated soft symbols v_k using (101). The timing error $\hat{\epsilon}_n$ is extracted by averaging the products of D_k and z_k over an update interval of K symbols(chips)

$$\Delta\epsilon_n = \frac{1}{K} \sum_{k=(n-1)K+1}^{nK} D_k \cdot z_k. \tag{110}$$

New timing reference $\hat{\epsilon}_{n+1}$ for symbols in the $(n+1)^{th}$ interval is estimated using the standard second order loop filter. Below shows the pseudo-code of our implementation of the DTTL loop.

Algorithm 4 DTTL

$\hat{\epsilon}_k \leftarrow \epsilon$ {Initialize timing estimate using saved state}
while input present **do**
  $n_\epsilon = \lfloor \hat{\epsilon}_n \cdot N_s \rfloor, \delta_\epsilon = \hat{\epsilon}_n \cdot N_s - n_\epsilon$ {Calculate parameters for the integrators}
  $v_k \leftarrow \sum_{i=(k-1)N_s+n_\epsilon+1}^{kN_s+n_\epsilon}((1-\delta_\epsilon)\cdot \mathrm{Im}(\tilde{y}_i) + \delta_\epsilon \cdot \mathrm{Im}(\tilde{y}_{i+1}))$ {In-phase integrator}
  $z_k \leftarrow C' \sum_{i=kN_s-\frac{n_w}{2}+n_\epsilon+1}^{kN_s+\frac{n_w}{2}+n_\epsilon}((1-\delta_\epsilon)\cdot \mathrm{Im}(\tilde{y}_i)+\delta_\epsilon \cdot \mathrm{Im}(\tilde{y}_{i+1}))$ {Mid-phase integrator}
  $D_k \leftarrow \frac{1}{2}(\mathrm{sgn}(v_{k-1}) - \mathrm{sgn}(v_k))$ {Data transition detection}
  $\Delta\hat{\epsilon}_n \leftarrow \frac{1}{K}\sum_{k=(n-1)K+1}^{nK} D_k \cdot z_k$ {Average timing error estimate}
  $\hat{\epsilon}_{n+1} \leftarrow \hat{\epsilon}_n + K_1\Delta\epsilon_n + K_2\sum_{m=1}^{n}\Delta\epsilon_m$ {Update timing reference estimate}
end while

The default value of W is 2^{-1} in the software, leading to a half symbol duration for the mid-phase integrator. Ideally W shall be large (e.g., $W = 1$) in the beginning of processing and gradually reduced as the timing is acquired.

D. Uplink PN Sequence Acquisition

Using the sequence of estimated soft symbols v_k, we can acquire the integer offset of the range clock as described in Section III-D. The functional block we implemented in the software is the same because it was already a discretized algorithm.

VI. Software Tools for Telemetry Ranging Support

In this section, we describe two useful software tools built from these functional blocks. One tool allows the user to simulate the end-to-end operation of a telemetry ranging system, allowing user inputs of system parameters to create various scenarios of interest. The second tool allows the software to directly process data recorded from the spacecraft and ground receivers of a telemetry ranging system in an actual or test configuration.

A. End-to-End System Simulation Tool

Fig. 19 shows a snapshot of the graphical user interface (GUI) we developed to facilitate end-to-end simulation and performance analysis. There are four control panels in the user interface that allow the user to configure the signaling format, channel condition, and receiver characteristics of the telemetry ranging system. We use the complex baseband signal model as described in Section III-A.

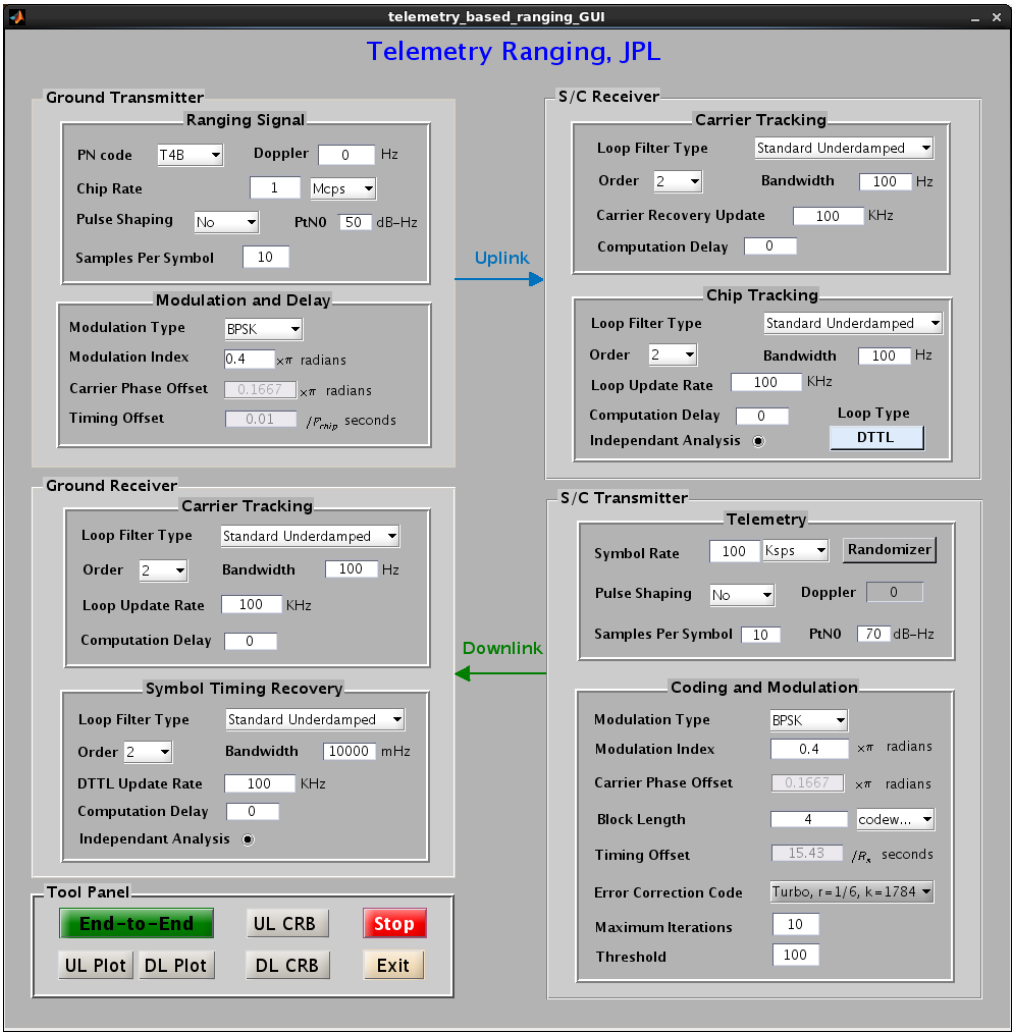


Figure 19. Graphical User Interface of the telemetry ranging software for end-to-end system simulations.

The ground transmitter module produces, in a revolving way, a noisy and delayed version of the uplink ranging signal $r(t)$. The PN code sequence d_k is chosen as one of the weighted-voting balanced Tausworthe codes (i.e., T2B or T4B). The drop-down menu for pulse shaping allows the user to select either a rectangular pulse shape or half-sine pulse shape based on the chip rate. In the simulations, the user specifies the number N_s of samples per chip, or digital symbol, and the sampling rate is set to N_s times the chip rate, where $N_s \geq 2$. The timing offset allows the user to set the simulated uplink delay $\tau_u(t)$ from the DSS to the spacecraft receiver. Even though this delay is set to be a constant in the user interface for simulation, our tracking loops are capable of tracking time-varying delays. The simulation sets the uplink instrumentation delays τ_u^{DSS}, τ_u^{SC} to zero.

The spacecraft receiver module provides options for the user to select appropriate loop

filter configurations for carrier tracking and chip tracking. The loop filter type can be chosen as either being "standard underdamped" or "supercritically damped". By default we always use 2nd-order loop filters. However the user can choose to use anything from 1st-order to 4th-order loops. The loop gain coefficients $\{K_i, i = 1, 2, 3, 4\}$ are completely characterized by the loop bandwidth B_L and update rate $1/T_u$ [11]. For chip timing tracking, the user can use either DTTL or DPLL by toggling the loop type button in the chip tracking control panel. The range code phase $\psi_s(t_S) = 2(\hat{s} + \hat{\epsilon})$ is determined from the acquired PN code phase s and chip timing offset ϵ as in (84).

The spacecraft transmitter module simulates generation of the downlink telemetry signal $s(t)$, as in Figure 3. The telemetry data is currently being generated as random sequences of $+1$s and -1s in the simulation software, without including the uplink range code phase measurements. For error correcting codes, the user is allowed to select from a group of rate-1/6 turbo codes and AR4JA codes with different lengths. The maximum iterations and threshold are used by the iterative decoder as the criteria to decide when to stop decoding. The Attached Sync Marker (ASM) bit pattern is chosen for the selected channel code according to the CCSDS recommended standard [9].

The ground receiver uses a DPLL to track the carrier phase of the downlink signal when a residual carrier is present, i.e., $\phi_d < \pi/2$, and uses Costas loop when the carrier is suppressed, i.e., $\phi_d = \pi/2$. DTTL is used for symbol synchronization of the random data modulation. The configuration parameters for the loop filters are the same as in the uplink tracking loops, although they may use different values.

To simulate telemetry ranging in a continuous, memory-efficient way, we generate the uplink and downlink signals in blocks and "stitch" them together in the signal processing of the receiver. The receiver module performs this stitching by saving the state of the tracking loops at the end of processing of each block, and using the saved state as the initial state for processing of the next block. Anytime during the simulation, the user may examine the status of the signal acquisition by reviewing statistics of the NCO outputs of the tracking loops. The transient as well as current states of the tracking loops are plotted, and an updated standard deviation of the tracking error is reported. Fig. 20 shows an example of realtime reporting of the downlink signal acquisition status. The user can also choose to use the "UL CRLB" and "DL CRLB" buttons in the tool panel to evaluate system performance analysis under different channel conditions and compare with Cramér-Rao bounds. The "Independent Analysis" button in the receiver panels allows the user to evaluate the performances in both a cascaded or stand-alone mode. The performance analysis is a useful tool for validating the software, as well as creating a capability to quickly test telemetry ranging estimation accuracy in various scenarios of interest. We report simulation results and error analysis in Section VII.

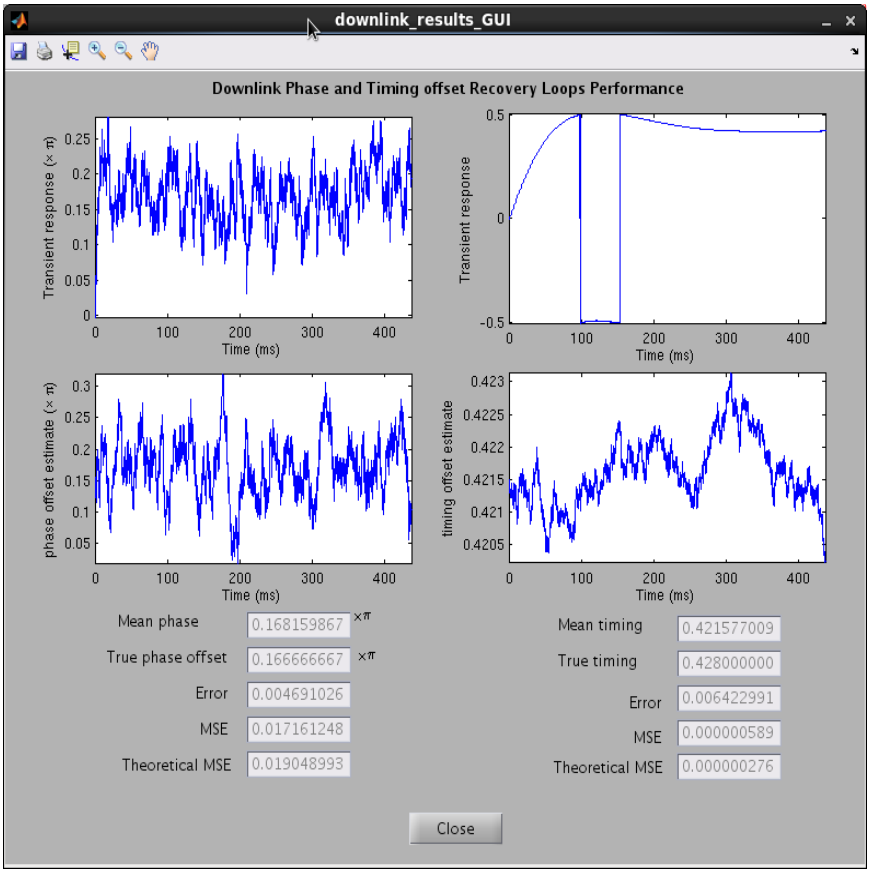


Figure 20. An example of realtime reporting of downlink signal acquisition status.

B. Test Data Processing Tool

In addition to the first tool enabling the user to run end-to-end system simulations, a second software tool allows direct processing of test data recorded from a telemetry ranging system. The GUI for this tool is shown in Figure 21. This tool provides the ground signal processing for receiving a telemetry-ranging signal recorded by a Radio Science Receiver (RSR), Very-Long Baseline Interferometry (VLBI) Science Receiver (VSR), or a Wideband VSR (WVSR). This tool was developed for testing at the Demonstration Test Facility 21 (DTF-21) in Monrovia, California. The GUI allows the user to select the open loop recorder type used and the name of the file containing the recording. All of the signal processing steps described in Section IV are carried out, culminating in a sequence of $\psi_s(t_s)$ phase measurements suitable for use in computing range.

VII. Performance Analysis and Numerical Results

This section starts with a brief derivation of Cramér-Rao bounds on the estimation performance of the individual major receiver functions on the spacecraft and the ground.

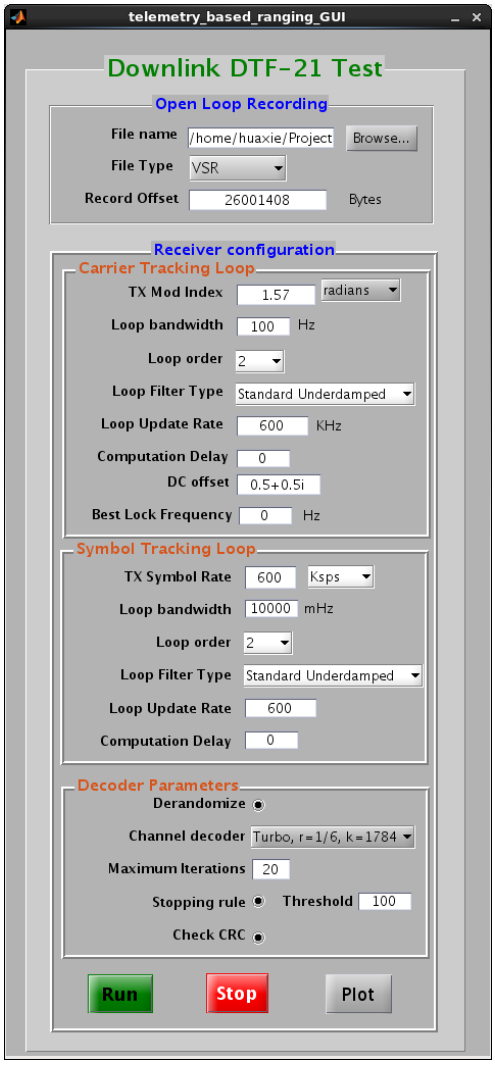


Figure 21. Graphical User Interface of the telemetry ranging software for processing test data.

These bounds are developed from continuous-time models of the corresponding processing stages, as in Section III and Section IV. Then we evaluate the simulated performance of the individual major receiver functions on the spacecraft and the ground in a standalone mode, isolated from errors in adjoining functional blocks, and compare these simulation results to the bounds. Then we evaluate the ranging estimation performance of the end-to-end telemetry ranging system and its component functional blocks in relevant scenarios.

A. Cramér-Rao Bound on the Variance of Estimation Error

The Cramér-Rao Bound (CRB) is a lower bound on the variance of any unbiased estimator, hence it provides a useful metric for evaluating the performance of the phase-locked loops and symbol synchronization systems used to estimate signal phase and symbol delay in this article.

When applied to telemetry ranging, the signal amplitude and noise variance may be assumed known, since these quantities remain essentially constant for a long time (up to hours), and even then change only slightly due to Earth rotation. Therefore, a long time is available to estimate these quantities accurately; we assume that these parameters are known for this application. However, the phase of the received carrier and the symbol or chip delay may drift on the time scale of seconds or minutes, due to residual Doppler caused by unmodeled relative velocity between the ground transmitter and the spacecraft. Therefore, these quantities must be estimated in real time to enable their measurement on the uplink and downlink signal as required for telemetry ranging.

The derivation of the CRB requires knowledge of the probability density of additive noise encountered during reception. It is therefore convenient to consider an N-vector of samples of the received signal plus noise for deriving the CRB, for which a joint probability density function can be constructed. It is assumed that the received radio-frequency signal has been downconverted to complex baseband and sampled in the presence of additive Gaussian noise, as described in (34) and (103). The received signal plus noise samples are modeled as $\tilde{r}_i = \tilde{s}_i + \tilde{n}_i$. The samples of the complex envelope of the downconverted signal are $\tilde{s}_i = \sqrt{P}\exp(j\theta)$, where $\sqrt{P}$ is the amplitude and θ is the unknown phase. The complex noise samples $\tilde{n}_i$ are zero-mean, Gaussian, and independent. Moreover, the real and imaginary components of each noise sample are independent and identically distributed.

For a vector of N independent complex noise samples, $\tilde{\mathbf{n}} = (\tilde{n}_0, \tilde{n}_1, \cdots, \tilde{n}_{N-1})$, the N-dimensional joint probability density $\mathfrak{p}(\cdot)$ is the product of the individual probability densities, given by

$$\mathfrak{p}(\tilde{\mathbf{n}}) = \left(2\pi\sigma_n^2\right)^{-N} \prod_{i=0}^{N-1} \exp\left(-\frac{|\tilde{n}_i|^2}{2\sigma_n^2}\right), \tag{111}$$

where $\sigma_n^2 = N_0/(2T_s)$ is the variance of each noise component. Since $\tilde{r}_i = \tilde{s}_i + \tilde{n}_i$, it follows that $\tilde{n}_i = \tilde{r}_i - \tilde{s}_i$ and we can equivalently write the natural logarithm of (111)

as

$$\Lambda(\tilde{\mathbf{r}}|\theta) = \ln \mathfrak{p}(\tilde{\mathbf{r}}|\theta) = -N \ln(2\pi\sigma_n^2) + \frac{\sqrt{P}}{\sigma_n^2} \operatorname{Re}\left\{ e^{-j\theta} \sum_{i=0}^{N-1} \tilde{r}_i \right\} - \frac{NP}{2\sigma_n^2} - \frac{1}{2\sigma_n^2} \sum_{i=0}^{N-1} |\tilde{r}_i|^2 , \tag{112}$$

where $\Lambda(\tilde{\mathbf{r}})$ is the log-likelihood function: it contains all of the information in the joint probability density, but in a more manageable form since the logarithm of a product of sample densities is transformed into the sum of their natural logarithm.

According to the CRB, the variance of any unbiased estimator must be at least as great as the inverse of the Fisher information, defined as $I(\theta) = -E\left\{\partial^2 \Lambda(\tilde{\mathbf{r}}|\theta)/\partial\theta^2\right\}$, where $E\{\cdot\}$ is the expectation operator. Carrying out the indicated operations yields

$$I(\theta) = -E\left\{ \frac{\partial^2 \Lambda(\tilde{\mathbf{r}}|\theta)}{\partial\theta^2} \right\} = -E\left\{ -\frac{\sqrt{P}}{\sigma_n^2} \operatorname{Re}\left\{ \exp(-j\theta) \sum_{i=0}^{N-1} \tilde{r}_i \right\} \right\} = \frac{NP}{\sigma_n^2}, \tag{113}$$

from which the CRB follows as

$$\sigma_{\Delta\theta}^2 \geq [I(\theta)]^{-1} = \frac{\sigma_n^2}{NP}, \tag{114}$$

where $\sigma_{\Delta\theta}^2$ is the variance of the phase error $\theta - \hat{\theta}$, P is the average power of the signal that contributes to the phase synchronization process, and N is the number of independent complex samples that are processed.

For our purposes, it is more convenient to express the CRB in terms of the closed-loop bandwidth B_L and the signal-to-noise spectral density ratio P/N_0. Recall that a phase-locked loop with closed-loop bandwidth B_L can be viewed as a short-term integrator with effective integration time $T_L = 1/(2B_L)$, hence the effective number of T_s-second samples ($T_s \ll T_L$) in a T_L-second time interval is $N = T_L/T_s = 1/(2B_L T_s)$. Approximating N as the number of T_s-second samples per loop integration time T_L, $N = T_L/T_s = (2B_L T_s)^{-1}$, we can now express the CRB in the following simple and convenient form:

$$\sigma_{\Delta\theta}^2 \geq \frac{N_0}{2T_s NP} = \frac{N_0(2B_L T_s)}{2T_s P} = \frac{B_L}{P/N_0}. \tag{115}$$

Note that the signal power P refers to the signal power relevant to the problem, and hence may be replaced by P_t, P_r, P_c or P_d in a particular application.

The CRB (115) can be put into another form:

$$\sigma_{\Delta\theta}^2 \geq \frac{1}{\rho}, \tag{116}$$

where ρ is the signal-to-noise ratio in the loop. For a simple loop, $\rho = P/(N_0 B_L)$. The form of (116) is more generally useful because it suggests lower bounds on performance for loops where the CRB is not strictly applicable. If the noise is not Gaussian and statistically independent from one sample to the next, then (111) and the remainder of the CRB derivation do not strictly hold. These approximations based on (116) often serve as tight lower bounds on performance for more general loops, as demonstrated by simulations.

A key observation is that as long as the phase error sensor is linear with slope equal to one when the phase error is near zero, the same CRB holds, except for some loss factors if not all of the signal power is utilized (as in PN DPLL), and except for some second-order effects that appear at very high SNR. In this way, we can treat the various phase and delay estimation loops within the same framework, and compare their performance on a common basis.

B. Performance of Residual Carrier Tracking with PLL or DPLL

At moderate to high loop signal-to-noise ratios, the closed-loop behavior of a phase-locked loop is well approximated by the linear loop model, characterized in discrete time by the closed-loop transfer function $H_d(j\omega)$, where ω is the phase advance (in radians) per loop update. This transfer function has a closed-loop bandwidth B_L. The performance of the loop depends on the extent to which noise and interference are tracked by the loop.

1. Transfer Function of Second-Order Loop

The transfer function of a second-order, discrete-time loop, expressed in terms of the gain coefficients K_1 and K_2 is [10]

$$H_d(j\omega) = \left. \frac{D(z) - (z-1)^2}{D(z)} \right|_{z=e^{j\omega}}, \tag{117}$$

where ω is the phase advance (in radians) per loop update, and

$$D(z) = (z-1)^2 + (z-1)K_1 + zK_2 = z^2 + z(K_1 + K_2 - 2) + (1 - K_1). \tag{118}$$

For a standard-underdamped loop, $K_1 = (8/3)\hat{B}_L T_u$ and $K_2 = K_1^2/2$ where $\hat{B}_L$ is the desired loop bandwidth and T_u is the loop update period. Combining (117) and (118) yields

$$H_d(j\omega) = \frac{e^{j\omega}(K_1 + K_2) - K_1}{e^{j2\omega} + (K_1 + K_2 - 2)e^{j\omega} + (1 - K_1)}. \tag{119}$$

The noise-equivalent bandwidth B_L of the loop is

$$B_L = \frac{1}{2\pi T_u} \int_0^{\pi} |H_d(j\omega)|^2 \, d\omega. \tag{120}$$

As long as the loop update rate is large enough, $1/T_u > 20\,\hat{B}_L$, the desired loop bandwidth is approximately achieved: $B_L \approx \hat{B}_L$.

The frequency response of the above loop may also be characterized as the transfer function of a continuous-time, equivalent system:

$$H(j2\pi f) = H_d(j2\pi f T_u). \tag{121}$$

In Figure 22, the $|H(j2\pi f)|^2$ defined by (119) and (121) is plotted, for the case $\hat{B}_L =$ 100 Hz, as a function of frequency f.

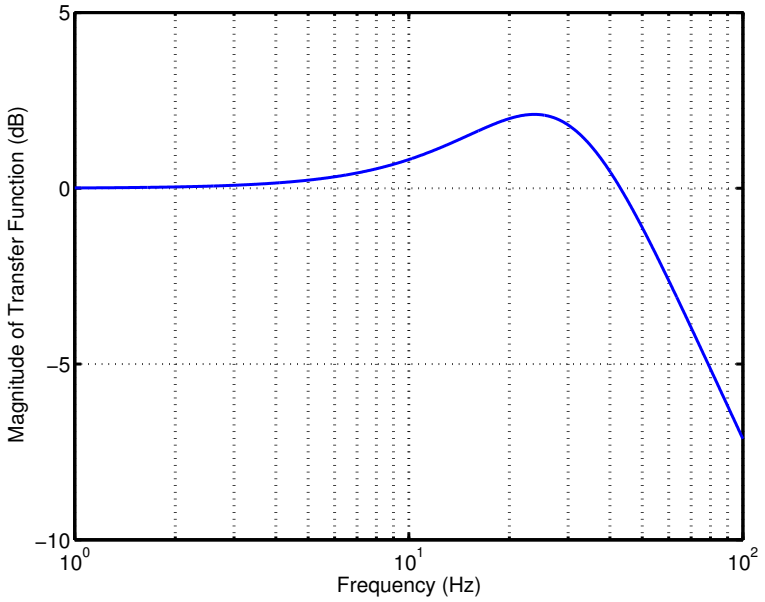


Figure 22. Squared magnitude of transfer function response of second-order, standard-underdamped DPLL with $B_L = 100$ Hz

The noise power P_n tracked by the loop is given by

$$P_n = N_0 B_L. \tag{122}$$

When a signal (range code or a data stream) modulates the carrier and a residual-carrier loop is used by the receiver, some of the low-frequency content of the modulation sidebands will lie inside the loop bandwidth, and this modulation can degrade the carrier-tracking performance. In order to account for this effect, we must first calculate the power P_Q on the Q channel in the receiver that enters the loop.

When the modulating signal consists of random data with rectangular pulses, such as BPSK, there is a simple formula for P_Q [12]:

$$P_Q = 2P_d T_d B_L, \tag{123}$$

where T_d is the period of the data symbol.

When the modulating signal is a range code, a numerical approach is required to calculate P_Q. This procedure is described here for the case of rectangular pulse shapes. In this case, the interfering modulation is $j\sqrt{P_r}e^{j\theta(t)}\phi'_{\mathrm{PN}}(t)$, as can be seen in (39). This term is in the Q channel of the receiver and so will interfere with the detection of the phase error signal in a residual-carrier tracking loop. We need to calculate the power spectrum of this term. Equivalently, we calculate the power spectrum of $\sqrt{P_r}\phi'_{\mathrm{PN}}(t)$.

The range code $\phi'_{\mathrm{PN}}(t)$ is (approximately) periodic and so its frequency content can be

found from a Fourier series expansion:

$$\phi'_{\text{PN}}(t) = \sum_{k=-\infty}^{\infty} A_k e^{-j2\pi kt/(LT)}, \tag{124}$$

where $L = 1009470$ (the number of chips in one period), T is the chip period, and the Fourier-series coefficients A_k are

$$A_k = \frac{1}{LT} \int_0^{LT} \phi'_{\text{PN}}(t)\, e^{-j2\pi kt/(LT)}\, dt. \tag{125}$$

The range code received at the transponder is not *exactly* periodic since the uplink delay will generally be time varying; however, the approximation of $\phi'_{\text{PN}}(t)$ as periodic is acceptable for the calculation of P_Q. In the case where the range code consists of rectangular pulses, The coefficients A_k can be written in terms of the L-point discrete Fourier transform $D_L(k)$ of the chip sequence $d_i = \pm 1$, $0 \le i \le L-1$:

$$A_k = e^{-j\pi k/L} \cdot \frac{\sin(\pi k/L)}{\pi k} \cdot D_L(k). \tag{126}$$

The powers $P_{\text{PN}}(k)$ in the discrete spectral lines of $\sqrt{P_r}\phi'_{\text{PN}}(t)$ are calculated as

$$P_{\text{PN}}(k) = P_r\, |A_k|^2 . \tag{127}$$

The frequency index k is a signed integer, so (127) represents a *two-sided* characterization of the power spectrum. The frequency corresponding to the frequency index k is $k/(LT)$. Conservation of energy is expressed by:

$$\sum_{k=-\infty}^{\infty} P_{\text{PN}}(k) = P_r. \tag{128}$$

Figure 23 shows the power spectrum of $\sqrt{P_r}\phi'_{\text{PN}}(t)$ as a function of the frequency offset from the residual carrier. This figure shows the case of the T4B range code with a 1.0-MHz chip rate and a ranging modulation index $\phi_r = 0.1\pi$. The vertical axis indicates the power of individual discrete spectral lines relative to P_t (the total power of the modulated carrier) in decibels. The largest spectral lines occur at the odd harmonics of the range clock. In the case of Figure 23, the range clock, which is one-half of the chip rate, equals 0.5 MHz. The residual-carrier power is indicated on this figure with a red marker. (P_c nearly equals P_t for the small modulation index of 0.1π.)

The spectrum of Figure 23 is filtered by the transfer function of the loop. It may be noted that this spectrum has a spectral notch near zero frequency offset, which results in less interference power from a T4B range code than from random BPSK. The filtered spectrum $|P_{\text{PN}}(k)|^2\ |H_d\,(j2\pi k/(LT))|^2$ is shown in Figure 24. This figure shows the case of the T4B range code with a 1.0-MHz chip rate and a ranging modulation index $\phi_r = 0.1\pi$. The effect of the 100-Hz loop bandwidth is evident. The spectrum is symmetric, since the modulation is real. The deep notch in the spectrum for frequencies less than 100 Hz is not visible in Figure 23 due to the horizontal scaling in that figure.

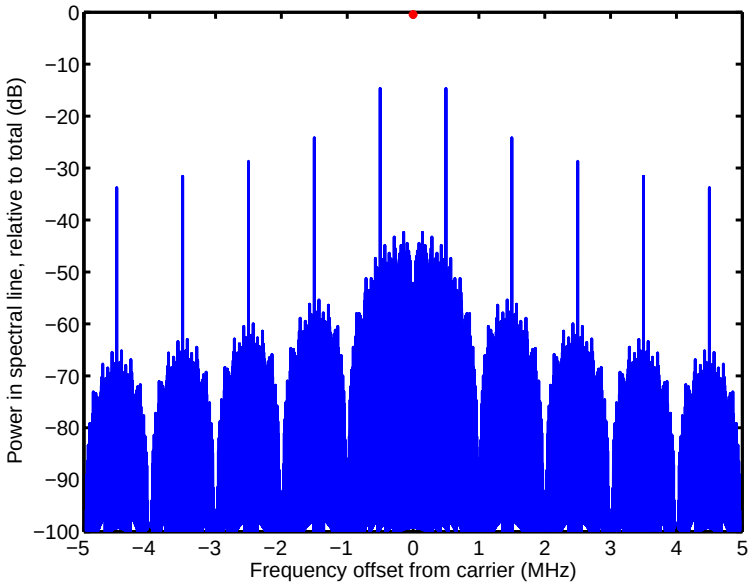


Figure 23. Spectrum of T4B range code modulation sidebands: 1 Mchips/s and $\phi_r = 0.1\pi$. Also, the residual-carrier power is shown with a red marker. (P_c is different from P_t by -0.4 dB in this case and so, because of the scale, appears to be 0 dBc.)

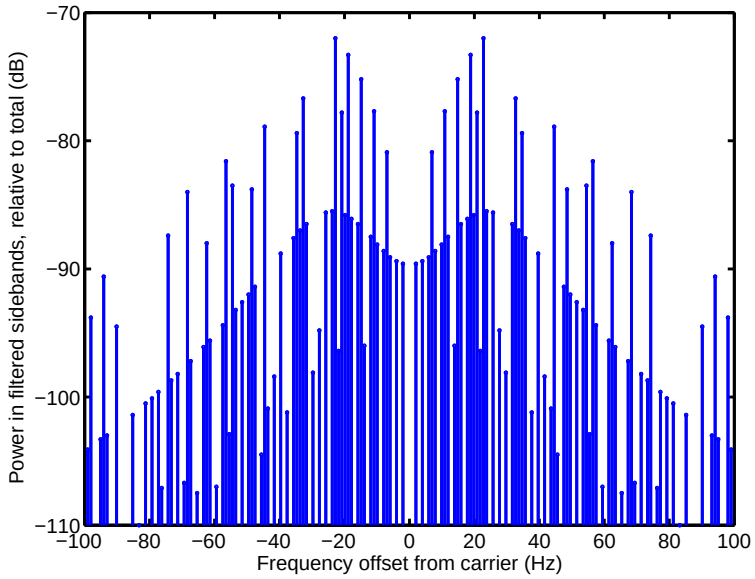


Figure 24. Filtered T4B PN code modulation spectrum: 1 Mchips/s and $\phi_r = 0.1\pi$.

The power P_Q is calculated as

$$P_Q = \sum_{k=-\infty}^{\infty} |P_{\text{PN}}(k)|^2 \, |H_d\left(j2\pi k/(LT)\right)|^2 . \tag{129}$$

2. Simulated Tracking Loop Performance

The performance of a residual-carrier tracking loop as determined through simulation is shown in Figure 25 as a function of P_c/N_0, for modulation indices of $\phi_r = 0.4\pi$, $0.1\pi/10$, and 0.01π, measured in terms of phase error standard deviation $\sigma_{\Delta\theta}$. The simulation results are represented by markers.

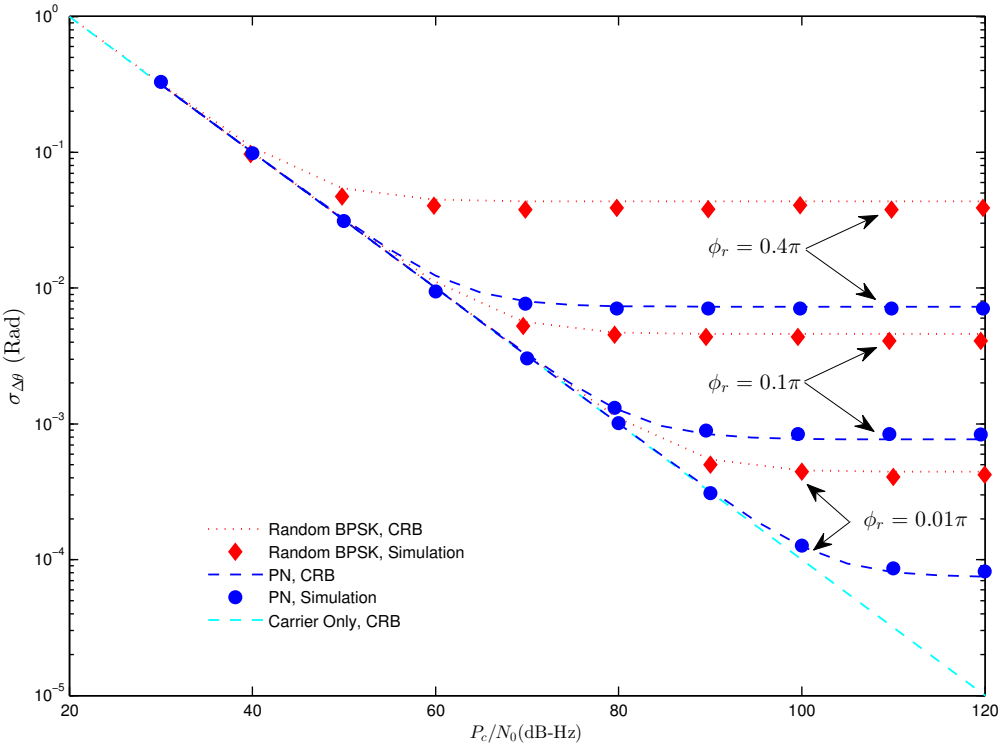


Figure 25. Phase error performance of DPLL carrier tracking loop as a function of P_c/N_0 (dB-Hz). Performance for both PN code and random data modulated signal are shown and compared with respective theoretical bounds. In the simulation, we used a 2nd-order DPLL with bandwidth $B_L = 100$ Hz. The chip rate and symbol rate is 10^6 bps, the loop filter update rate was set to 10^5 Hz.

We use (116) as an analytical model for the performance of the loop. In this model, the loop SNR ρ is

$$\rho = \frac{P_c}{P_n + P_Q}, \tag{130}$$

where P_n is given in (122) and P_Q is given in (123) or (129), depending the modulating signal is random or the T4B range code. Thus,

$$\sigma^2_{\Delta\theta} \geq \frac{P_n + P_Q}{P_c}. \tag{131}$$

This model is an approximation based on the CRB. Because the modulating signal does not have a Gaussian distribution, the CRB is not strictly applicable when $P_Q \neq 0$. The analytical models are shown in Figure 25 as curves.

The performance floor that is seen on the right-hand side of each curve in Figure 25 is due to the dominance of P_Q over P_n for large values of P_c/N_0. The floor is higher when the modulation index is larger.

Figure 25 shows that the analytical model agrees well with the simulation results. The analytical model even closely predicts the performance floor. The agreement is good over a wide range of values for ϕ_r and P_c/N_0.

C. Performance of Suppressed Carrier Tracking with a Costas Loop

From (95) and Figure 11, and noting that $P_d = P_t$ for a suppressed carrier signal, it can be seen that the input to the Costas loop filter is

$$-\frac{z_k}{P_d} \approx \Delta\theta_k - \frac{n_k''}{P_d}. \tag{132}$$

This can be compared to the input to the PLL loop filter, given by (58),

$$\frac{z(t)}{P_c} \approx \Delta\theta_k - \frac{n_s''(t)}{P_c}. \tag{133}$$

Since signal models for the PLL and Costas loops have the same form, the performance analysis for the PLL can also be used for the Costas loop. The difference is that the noise $n_s''(t)$ entering the PLL loop filter is Gaussian, whereas the noise n_k'' entering the Costas loop filter noise is given by (96), which we repeat here for convenience:

$$n_k'' \triangleq \sqrt{P_t}\cos(\Delta\theta_k)d_k n_{c,k}' - \sqrt{P_t}\sin(\Delta\theta_k)d_k n_{s,k}' + n_{c,k}' n_{s,k}'. \tag{134}$$

This includes two signal-noise cross terms and a Gaussian-times-Gaussian term. To describe the performance, we consider the low SNR and high SNR cases separately.

In the high SNR case, the Gaussian product term $n_{c,k}' n_{s,k}'$ in (134) may be ignored because it is dominated by the signal-noise cross terms. What remains is not Gaussian, but because a large number of such samples contribute to the filtered error signal, by the central limit theorem it may be considered approximately Gaussian. Thus, performance will be determined by the variance

$$\begin{aligned}
\text{var}\left[\frac{n_k''}{P_d}\right] &\approx \frac{1}{P_d^2}E[P_d\cos^2(\Delta\theta_k)d_k^2 n_{c,k}'^2 + P_d\sin^2(\Delta\theta_k)d_k^2 n_{s,k}'^2 \\
&\quad + P_d\sin(\Delta\theta_k)\cos(\Delta\theta_k)d_k^2 n_{c,k}' n_{s,k}'] \qquad (135)\\
&= \frac{1}{P_d}(\cos^2(\Delta\theta_k)\sigma^2 + \sin^2(\Delta\theta_k)\sigma^2) \qquad (136)\\
&= \sigma^2/P_d, \qquad (137)
\end{aligned}$$

where (136) follows because $n'_{c,k}$ and $n'_{s,k}$ are independent, each with variance $\sigma^2 = N_0 B_{\text{L}} = P_n$. Thus, in the high SNR case, the loop SNR is $\rho = P_d/P_n$, or

$$\sigma^2_{\Delta\theta} \geq \frac{P_n}{P_d} = \frac{N_0 B_{\text{L}}}{P_d} = \frac{B_{\text{L}}}{P_d/N_0}. \tag{138}$$

It can be seen that for high SNR the CRB is inversely proportional to P_d/N_0, and in fact it is equal to the CRB given in (131) for the residual carrier loop when modulation is not present.

In the low SNR case, the signal cross terms in (134) may be ignored because they are dominated by the noise product term. The product of two Gaussian random variables is not Gaussian, but because a large number of such samples contribute to the filtered error signal, by the central limit theorem it may be considered approximately Gaussian. In this case we have

$$\text{var}\left[\frac{n''_k}{P_d}\right] \approx E\left[\frac{n'^2_{c,k} n'^2_{s,k}}{P_d^2}\right] \tag{139}$$

$$= \frac{1}{P_d^2} E\left[n'^2_{c,k}\right] E\left[n'^2_{s,k}\right] \tag{140}$$

$$= \sigma^4/P_d^2, \tag{141}$$

where (140) follows by the independence of $n'_{c,k}$ and $n'_{s,k}$.

In the following development it is convenient to work with the samples z_k of the discrete model as shown in Figure 17, which now represent an average of N_s samples over the symbol duration. The variance of the averaged noise samples is lower than the variance of the individual samples by a factor of N_s, yielding the symbol-averaged noise variance $\sigma_z^2 = \sigma^2/N_s$. The normalized product samples $x_k \triangleq -z_k/P_t = \Delta\theta_k - n''_k/P_t$ contain the required error signal, as well as noise samples with variance $\sigma_x^2 = \sigma^4/(N_s^2 P_t^2)$. Recalling from [11] that the sample noise variance can be expressed in terms of the sample integration time T_s, signal power to noise spectral level P_d/N_0 as $\sigma^2 = P_t/(2T_s P_d/N_0)$, and with symbol duration $T = N_s T_s$, it follows that

$$\sigma_x^2 = \frac{\sigma^4}{N_s^2 P_t^2} = \frac{P_t^2}{4N_s^2 T_s^2 (P_d/N_0)^2 P_t^2} = \frac{1}{4T^2(P_d/N_0)^2}. \tag{142}$$

Letting N_c be the number of T second symbol times in an inverse loop bandwidth, $N_c = 1/(2B_L T)$, and substituting σ_x^2 into (114) and noting that x has been normalized to have unit power, yields $\sigma^2_{\Delta\theta} \geq \sigma_x^2/N_c$, or in terms of T, N_c, and P_d/N_0,

$$\sigma^2_{\Delta\theta} \geq \frac{\sigma_x^2}{N_c} = \frac{1}{4N_c T^2 (P_d/N_0)^2} = \frac{B_L}{2T(P_d/N_0)^2}. \tag{143}$$

This expression is valid for the limiting case when the signal-noise cross terms dominate. It can be seen that the CRB is inversely proportional to the square of P_d/N_0 for the low SNR case.

The Costas loop has been simulated over a range of P_d/N_0 that covers both the low SNR and high SNR regions, and its performance compared with the CRBs derived above:

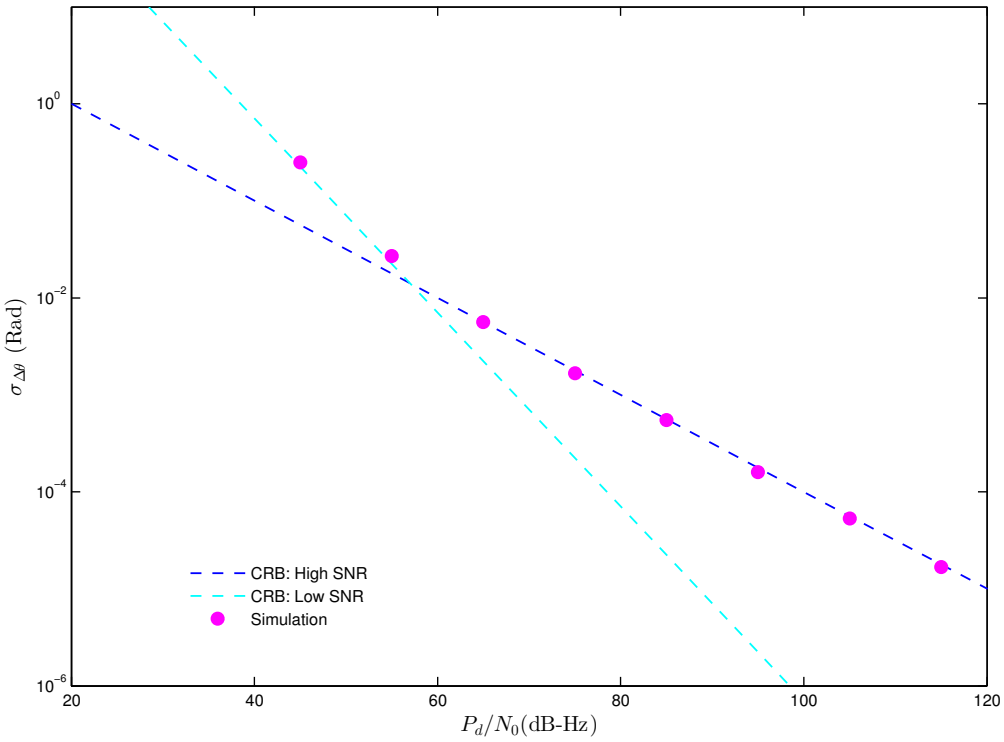


Figure 26. Carrier tracking performance of Costas loop as a function of P_d/N_0. Simulation points were produced using random data with symbol rate $R_s = 1$ Msps, loop filter bandwidth $B_L = 100$ Hz, and loop update rate $R_u = 10^5$ Hz.

(138) and (143). It can be seen in Figure 26 that the CRB expression in (143) is accurate in the low SNR region of $P_d/N_0 < 50$ dB-Hz, where $\sigma_{\Delta\theta} \propto P_d/N_0$, and likewise (138) is accurate in the high SNR region of $P_d/N_0 > 60$ dB-Hz, where $\sigma_{\Delta\theta} \propto \sqrt{P_d/N_0}$. The cross-over occurs when

$$\frac{B_{\text{L}}}{P_d/N_0} = \frac{B_L}{2T(P_d/N_0)^2}, \tag{144}$$

or $P_d/N_0 = 1/(2T) = R_s/2$. For the example shown in Figure 26 with $R_s = 1$ Msps, the cross over occurs at $P_d/N_0 = 57$ dB-Hz.

D. Performance of DTTL for Downlink Data Tracking

For a conventional DTTL, the equation characterizing loop error is [12]

$$\sigma_\tau^2 = \frac{W B_L}{2S \cdot P_d/N_0} \quad \text{chips}^2. \tag{145}$$

where the squaring loss is given by

$$S = \frac{\left[\text{erf}\left(\sqrt{E_S/N_0}\right) - \frac{W}{2}\sqrt{\frac{E_S/N_0}{\pi}}\, e^{-E_S/N_0}\right]^2}{1 + \frac{W}{2}\, E_S/N_0 - \frac{W}{2}\left[\frac{1}{\sqrt{\pi}} e^{-E_S/N_0} + \sqrt{E_S/N_0}\,\text{erf}\left(\sqrt{E_S/N_0}\right)\right]^2}, \tag{146}$$

where

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2}\, dy. \tag{147}$$

The "stand-alone" (i.e., with perfect carrier tracking of the input signal) performance of the DTTL is shown in Figure 27, where the fractional delay error was converted to range error by noting that $T_c = 10^{-6}$ seconds delay is equal to 300 meters of range at the speed of light: the simulation points (red circles) follow the high-SNR CRB closely above 53 dB-Hz. Squaring loss is also illustrated in Figure 27, for $P_t/N_0 < 53$ dB-Hz, which is considered to be the low-SNR region, with a tracking threshold evident at 47 dB-Hz, below which the DTTL loses lock.

E. Performance of Range Clock Tracking with DPLL or DTTL

It is possible to track the range clock with a DPLL, owing to the strong range-clock component within the range code. The performance of DPLL tracking of the range code is discussed below. Following that is a comparison of the DPLL and the DTTL for tracking the range code.

1. DPLL Tracking of PN Chip Timing

Within the transponder, the baseband range-code signal that results from carrier demodulation is described in (73). The range code has a signal coefficient $2\sqrt{P_r}/\pi$, and the Q-channel has a noise spectral density (two-sided) of $N_0/4$.

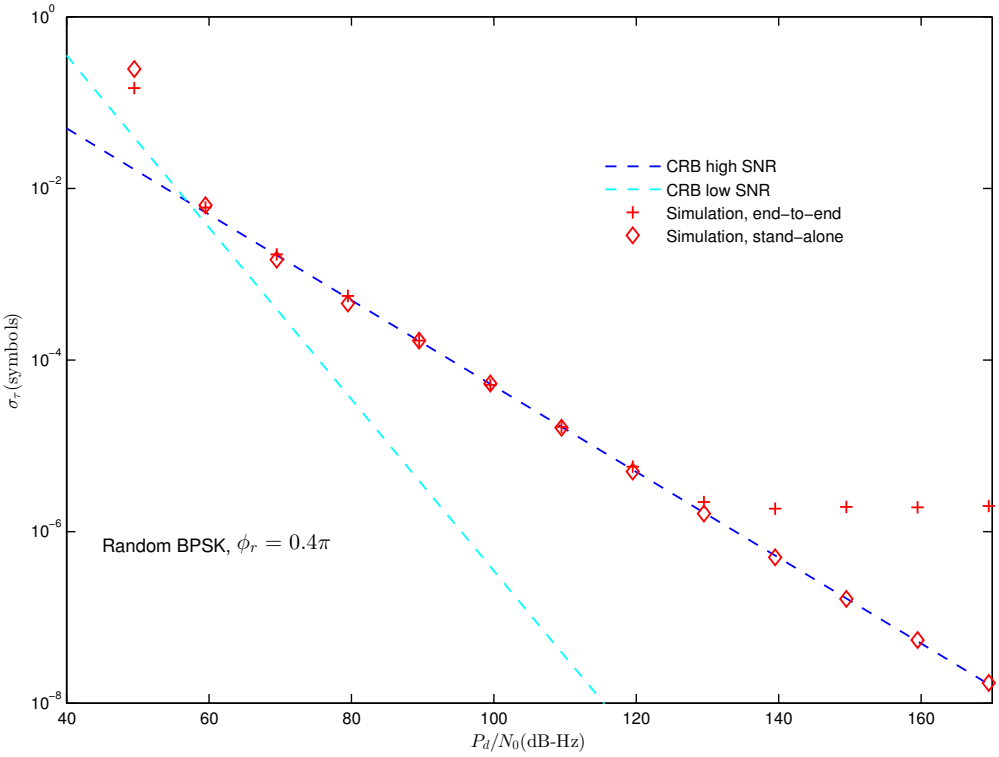


Figure 27. Standard deviation of symbol tracking error using DTTL. Performance was evaluated both with (end-to-end) and without (stand-alone) preceding carrier tracking loop jitter. Random BPSK data modulation at 1 Msps was used with $\phi_d = 0.4\pi$. The loop filter bandwidth was $B_L = 100$ Hz, and loop update rate $R_u = 10^5$ Hz.

After normalizing by the signal coefficient, the phase error input to the NCO for correction is $\Delta\theta_{\text{RC}} = \theta_{\text{RC}} - \hat{\theta}_{\text{RC}}$. If the range code phase is constant and the filtered noise is negligibly small, then $\Delta\theta_{\text{RC}} \simeq 0$ and the loops estimate of the range-code phase is accurate: $\hat{\theta}_{\text{RC}} \simeq \theta_{\text{RC}}$. Note that there is no contamination from the residual phase error of the carrier tracking loop, hence this estimate represents the range-code phase, as required for Telemetry Ranging applications. The accuracy of the estimate depends on the signal power and noise spectral level, as the following derivation demonstrates.

For a ranging signal with rectangular pulse shape, the SNR in the range-tracking DPLL is

$$\rho = \frac{8P_r}{\pi^2 B_{\text{L}} N_0}, \tag{148}$$

where B_{L} is the loop bandwidth of the range-tracking DPLL. From (116), the CRB is

$$\sigma^2_{\Delta\theta_{\text{RC}}} \geq \frac{1}{\rho} = \frac{\pi^2 B_{\text{L}}}{8P_r/N_0} \quad \text{radians}^2, \tag{149}$$

or, since 2π radians corresponds to two chips,

$$\sigma^2_{\Delta\theta_{\text{RC}}} \geq \frac{1}{\rho} = \frac{B_{\text{L}}}{8P_r/N_0} \quad \text{chips}^2. \tag{150}$$

However, this does not take into account the fact that the range code is not a perfect square-wave, but contains occasional inversions of the C_1 component, leading to a power loss factor of approximately 0.9 for the T4B code. This loss factor should be incorporated into the signal power, for a more accurate bound.

We note the importance of taking the imaginary part of $\tilde{y}(t)$ prior to mixing with the range-clock frequency. If this had not been done, it is straightforward to show that

$$\tilde{y}(t) \cdot \exp[-j2\pi f_{\text{RC}} t] \approx \frac{2\sqrt{P_r}}{\pi} \exp[j(\Delta\theta(t) + \theta_{\text{RC}})] + n'(t), \tag{151}$$

which would cause tracking performance to be degraded whenever $\Delta\theta(t) \neq 0$, even when $\Delta\theta(t) \ll 1$. By including only the imaginary component of $\tilde{y}(t)$, the performance is virtually independent of $\Delta\theta(t)$, provided $\Delta\theta(t)$ varies much slower than f_{RC} so that the approximation in (71) is accurate.

This loss factor has been incorporated into the CRB to obtain the performance of the DPLL, modified to track compound PN codes that can be well approximated by a square-wave. The results are shown in Figure 28, over a nominal operational range of 40–180 dB-Hz.

It can be seen in Figure 28 that the simulation results follow the CRB over a wide range of SNR, namely $40 < P_r/N_0 < 110$ dB-Hz, encompassing the typical operating range of 50 to 60 dB-Hz commonly employed in the DSN. However, at higher values of P_r/N_0, a flooring effect on the rms phase error can be observed, as shown in Figure 28. The red circles are simulation results.

This flooring is due to the fact that the product of the square-pulse compound PN code with the complex exponential yields a high-frequency process which is eventually

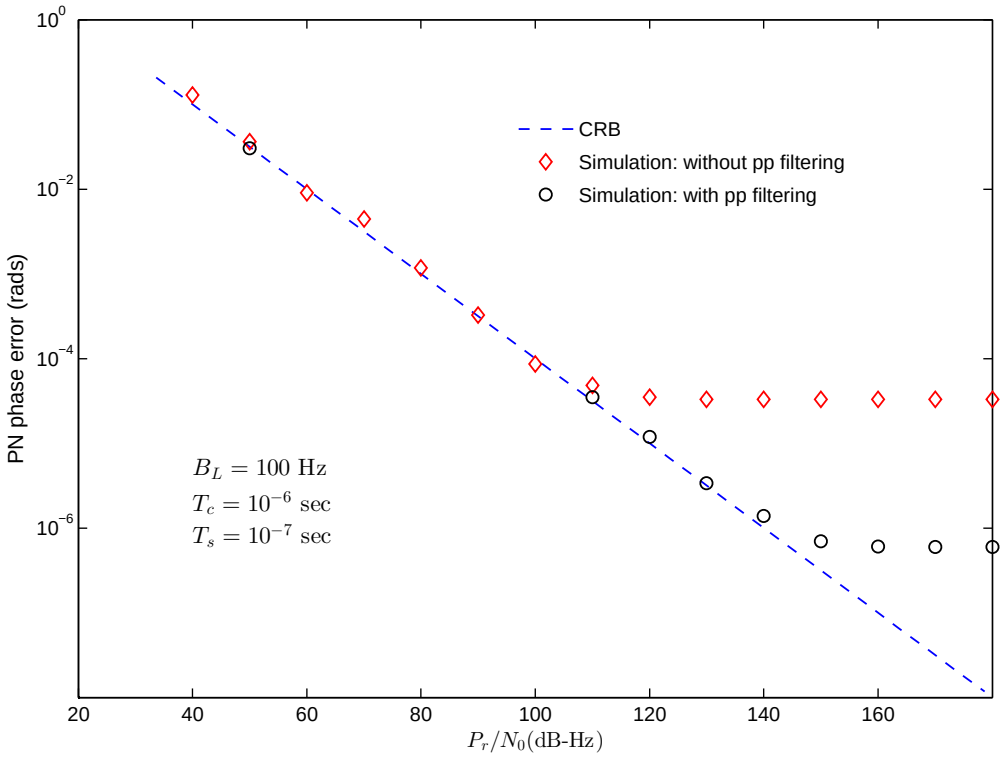


Figure 28. Performance of chip tracking loop as a function of P_r/N_0, over the nominal operating range of P_r/N_0 = 40–180 dB-Hz. $B_{\mathrm{LP}} = 500$ Hz.

filtered but not completely eliminated by the loop with closed-loop bandwidth B_{L}. After taking the imaginary component of the product, we have $\left|\sqrt{P_r}\phi(t)\sin(2\pi f_{\text{RC}}t)\right|$, when the compound PN code $\phi(t)$ is a square-wave function (a good approximation to the T4B code). An example of the residual filtered product waveform is shown in Figure 29(e) in the absence of additive noise ($P_r/N_0 = \infty$): the zoomed version in Figure 29(f) shows the residual product waveform, including the occasional glitches characteristic of compound PN codes, which begins to dominate performance above 110 dB-Hz.

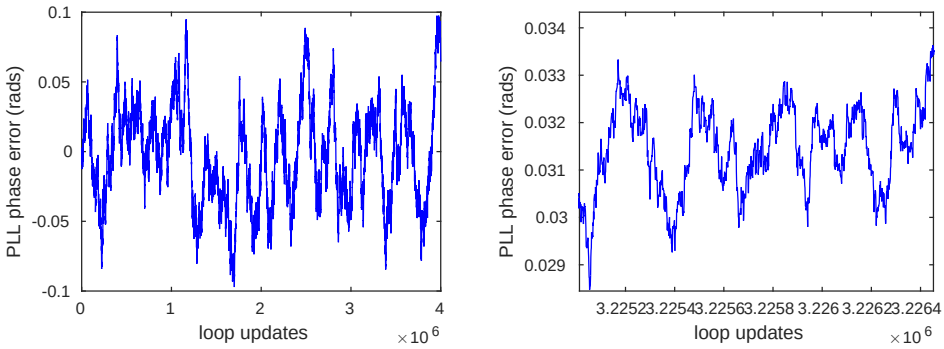
At lower SNR, the additive noise dominates, whereas the rms value of the additive noise is comparable to the residual product term near the start of the floor (110 dB-Hz), as illustrated by the phase error trajectories of Figures 29(a)-(d)

The residual product component has been ignored in the above analysis, since it is negligible at most operating points of interest, however at very high SNR it becomes the dominant component of the phase error if this product-term is allowed to enter the tracking loop. Most of this high-frequency product component can be eliminated by placing a properly designed low-pass filter after the product but before the loop. The bandwidth of this low-pass filter is denoted B_{LP}.

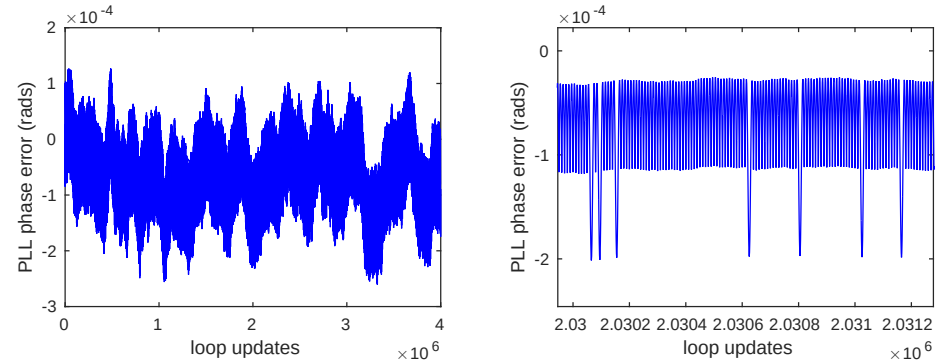
The constraints on the bandwidth B_{LP} of this filter are that it must be significantly greater than the close-loop bandwidth B_{L}, so as not to change the design bandwidth, but much less than the chip bandwidth B_c in order to filter it effectively: $B_{\text{L}} < B_{\text{LP}} < B_c$. For the performance plot of Figure 28 the low-pass post-product filter bandwidth was set to 500 Hz.

The impact of the 500 Hz post-product filter on the phase error trajectory of the PN tracking loop can be seen in Figure 30.

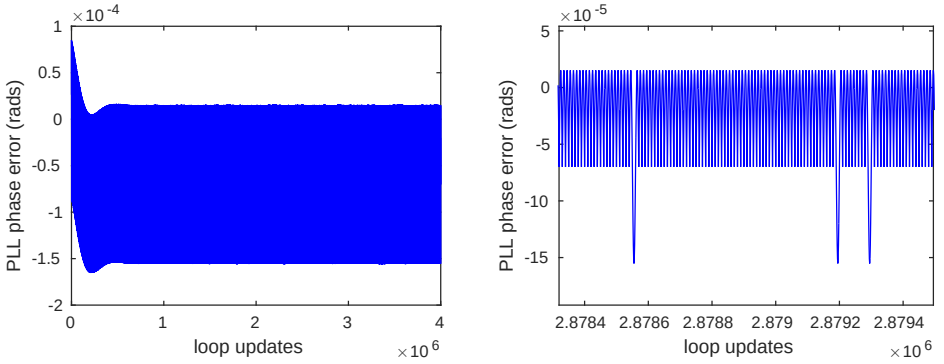
With this filter, additive noise dominates at both low SNR (about 30 dB-Hz) and even at 110 dB-Hz, where flooring begins to occur without post-product filtering, as seen in Figures 30(a)-(d). A small component of the product waveform remains even in the absence of additive noise, seen in Figures 30(e)-(f), however the phase error is dominated by another phenomenon that appears to be random but is not due to random noise (since no noise was added in the simulation). The cause for this fluctuation is the uneven distribution of "glitches" or negative-going spikes that introduce an irreducible phase noise into the output. Since the DPLL tracks the fundamental sinusoidal component of the PN code, the number and distribution of glitches over a closed-loop coherence time (the inverse of the closed-loop bandwidth) fluctuates as the loop filters the incoming signal, effectively changing the phase of the fundamental Fourier component. This fluctuation is due entirely to the structure of the PN code, and hence the phase error signature remains the same from one simulation run to the next, provided the loop bandwidth is fixed. This irreducible fluctuation of the phase of the fundamental Fourier component creates another, much lower phase error floor starting at 150 dB-Hz, of approximately 5×10^{-7} radians rms, as can be seen in Figure 28 (black circles). With chips, the ultimate limit on ranging accuracy imposed by this irreducible limit is (600



(a) $P_d/N_0 = 50$ **dB-Hz** **(b)** $P_d/N_0 = 50$ **dB-Hz, zoomed**

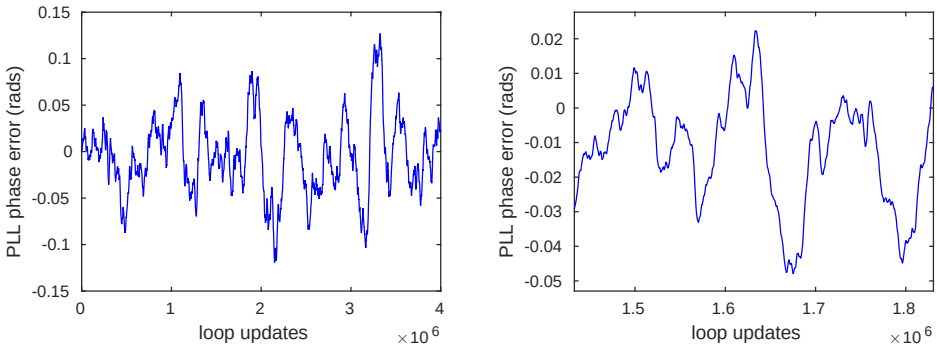


(c) $P_d/N_0 = 110$ **dB-Hz** **(d)** $P_d/N_0 = 110$ **dB-Hz, zoomed**

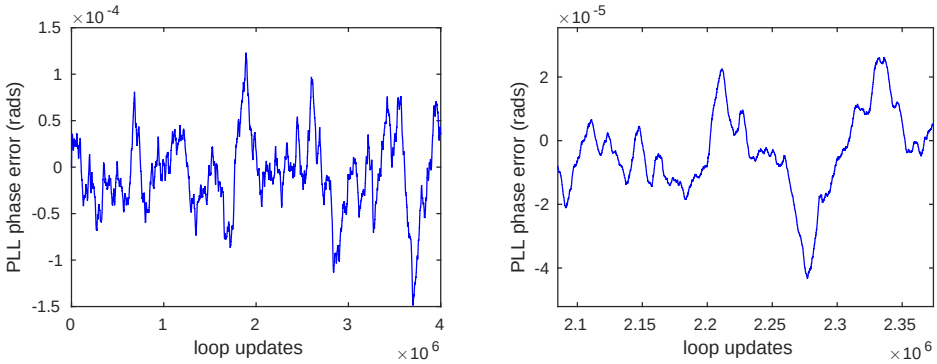


(e) $P_d/N_0 = \infty$ **dB-Hz** **(f)** $P_d/N_0 = \infty$ **dB-Hz, zoomed**

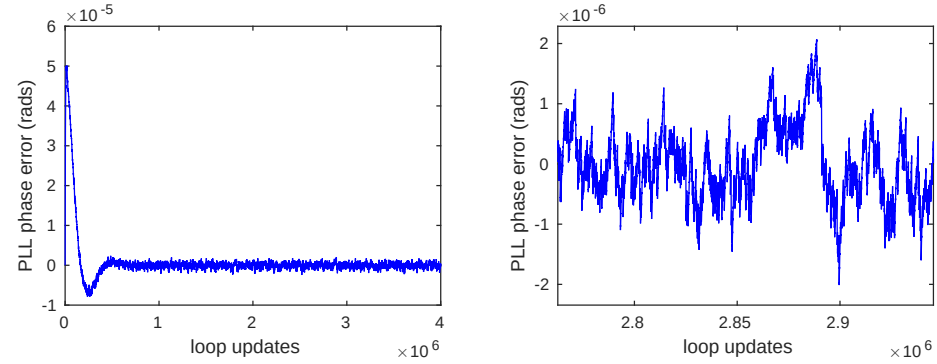
Figure 29. Phase error trajectories of DPLL when tracking PN code T4B, without post-product filtering.



(a) $P_d/N_0 = 50$ **dB-Hz** **(b)** $P_d/N_0 = 50$ **dB-Hz, zoomed**



(c) $P_d/N_0 = 110$ **dB-Hz** **(d)** $P_d/N_0 = 110$ **dB-Hz, zoomed**



(e) $P_d/N_0 = \infty$ **dB-Hz** **(f)** $P_d/N_0 = \infty$ **dB-Hz, zoomed**

Figure 30. Phase error trajectories of DPLL when tracking PN code T4B, with 500 Hz post-product filtering.

meters/cycle)$(5 \times 10^{-7}) = 3 \times 10^{-4}$ meters, or 0.3 mm.

2. Comparison of DPLL and DTTL Tracking of PN Chip Timing

Figure 31 compares the performances of DPLL and DTTL in terms of chip tracking jitter. This comparison was performed with the presence of a preceding carrier tracking jitter. Note that when DPLL is used, only the imaginary component of the carrier tracking output is pre-multiplied with a sinusoidal waveform before entering the DPLL tracking loop, which leads to a performance advantage. Comparing (150) to (145), one

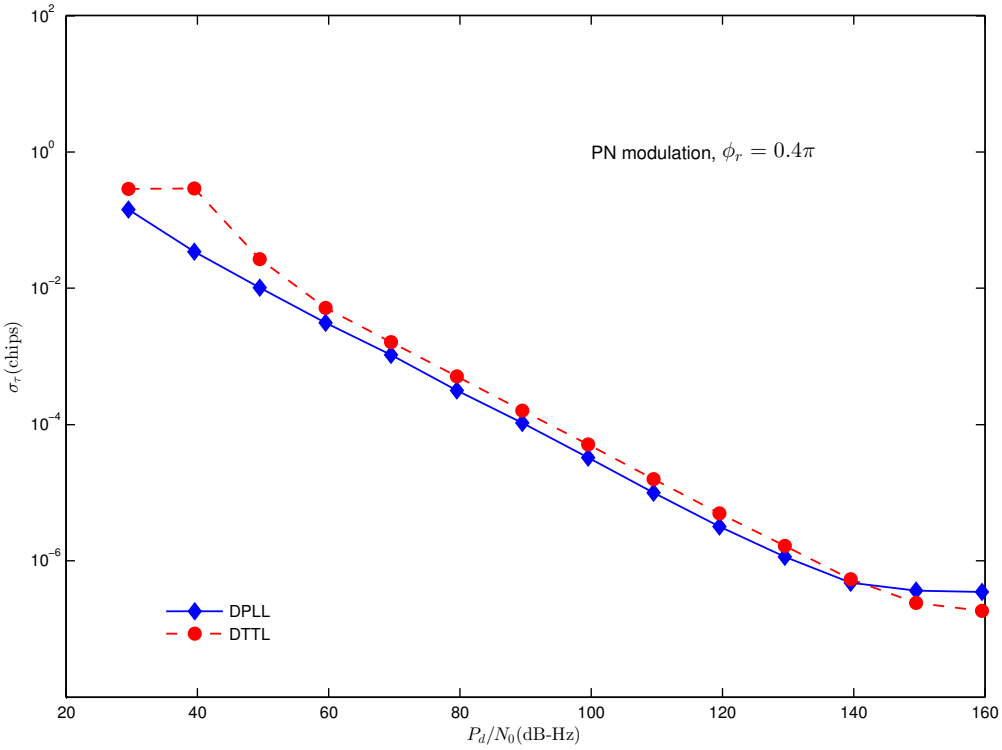


Figure 31. Comparison of chip tracking performances of DPLL and DTTL. Modulation index $\phi_r = 0.4\pi$, loop filter bandwidth $B_L = 100$ Hz, and $R_c = 10^6$ chips/sec, and update rate $R_u = 10^5$ Hz.

can see that the DPLL is better than the DTTL by a factor of $4W/S$ in its variance. Since the DTTL in this example used $W = 1/2$, the DPLL variance is smaller than the DTTL variance by a factor of two, for SNRs that have small squaring loss. The $\sqrt{2}$ difference between the DPLL and DTTL standard deviation can be seen in Figure 31, for $60 < P_d/N_0 < 130$ dB-Hz.

F. Performance of Cascaded Performance of Carrier Tracking and Symbol Tracking Loops

When the residual phase error from the carrier tracking loop cannot be ignored, then the impact of the phase error term on DTTL performance must be considered. If the carrier is not fully suppressed and the DTTL is preceded by a carrier tracking loop that suffers

from flooring due to modulation interference, the residual phase error from the carrier tracking loop appears in the NCO signal, as the following argument demonstrates. Since the chip rate is typically orders of magnitude greater than the loop bandwidth, the residual phase error process can be considered constant over several chip intervals, adding in a constant offset to both the in-phase and quadrature integrals. A slowly varying offset does not impact the transition detection statistic because the second in-phase integral is subtracted from the first, hence the common offset is nulled by the transition detector. However, the quadrature (or mid-phase) statistic contains the integral of the residual phase over the quadrature interval, which then adds directly to the error signal. Therefore, if the NCO output is used to evaluate DTTL performance, a flooring effect may be observed at high SNR, due to flooring in the carrier tracking loop when modulation interference is significant.

Figure 32 shows the chip timing accuracy using DPLL loop. The performance is evaluated both with and without the presence of the carrier tracking jitter. The simulation results agree really well with analytical approximations based on the CRB. The modulation index ϕ_r was 0.4π for this experiment.

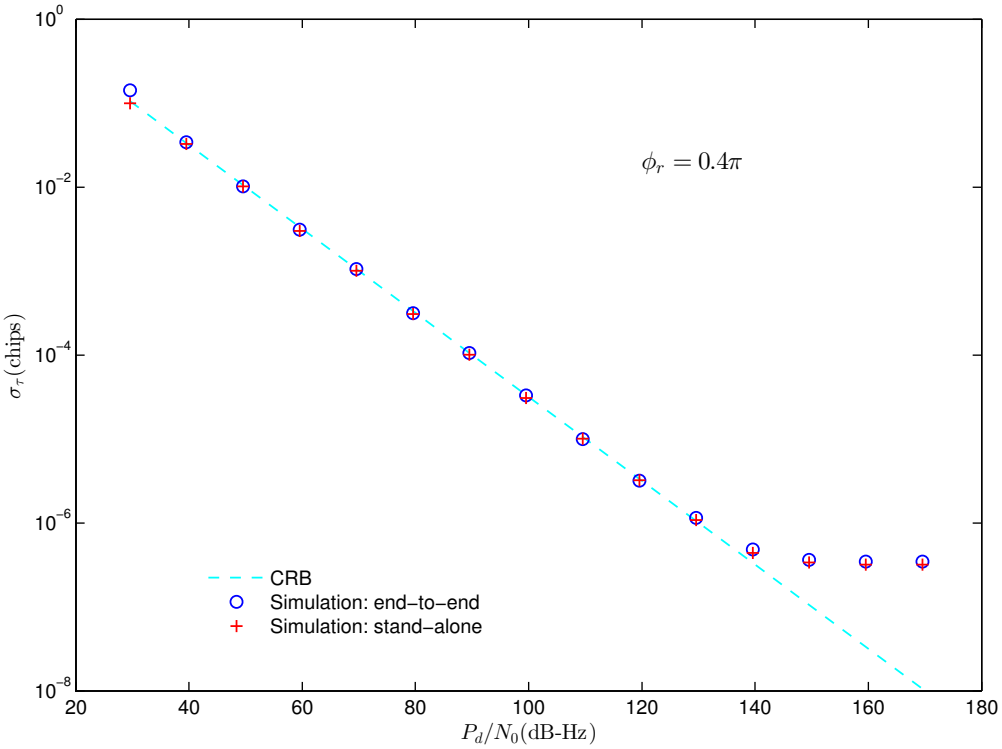


Figure 32. Chip tracking performance using DPLL with loop filter bandwidth $B_L = 100$ Hz and update rate $R_u = 10^5$ Hz. Tracking performance with (end-to-end) and without (stand-alone) presence of carrier tracking jitter are compared with CRBs.

Examples of the performance of a system consisting of a cascaded carrier tracking loop followed by a DTTL symbol synchronizer are shown in Figure 33, illustrating the flooring

effect in the NCO output of the DTTL when $\phi_r = 0.4\pi$, for two cases of interest: a) when the flooring is due to interference from a PN code, as would be observed at the spacecraft in the uplink leg of the telemetry ranging application; and b) when the flooring is due to random data received on the ground, if a residual carrier is used on the downlink.

In the simulation results shown in Figure 33, the loop bandwidth was set to 100 Hz for both the carrier tracking loop and the DTTL: the chip rate was set to 10^6 chips per second, with 10 samples per chip, and the loop update rate was set to 10^5. It is apparent in Figure 33, that flooring occurs at approximately 20 dB lower P_d/N_0 with random data than with the PN code, and that flooring due to random data is approximately a factor of ten greater when measured in terms of fractional chip errors, than for PN codes. This is attributed to the fact that the spectral level of random data is flat in the frequency region of the closed-loop bandwidth, whereas the spectrum of the PN code has a significant notch at low frequencies, thus contributes less interference to the loop.

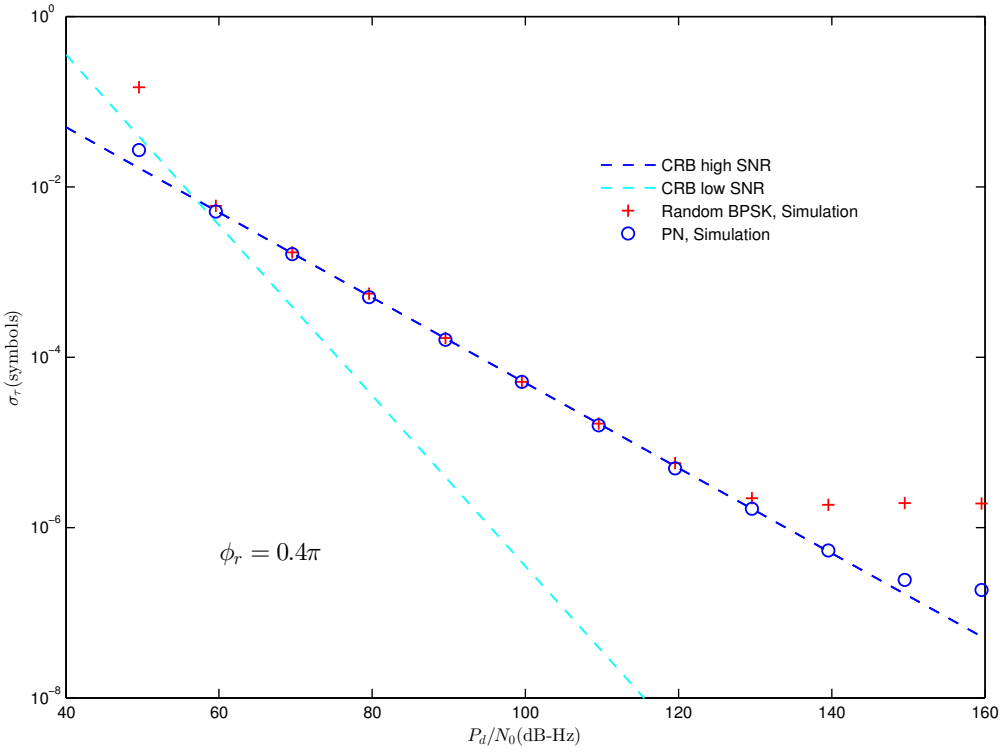


Figure 33. Performance of cascaded carrier tracking loop (DPLL) followed by a DTTL for both PN code and random data modulation with $\phi_{r(d)} = 0.4\pi$. The chip rate or symbol rate was set to 1 Mbps, loop filter bandwidth $B_L = 100$ Hz , and loop update rate $R_u = 10^5$ Hz. These parameters are the same for both DPLL and DTTL in the cascaded system.

VIII. Conclusions

In this article we described the operation of the major receiver functions on the spacecraft and the ground that are required to support telemetry ranging. Many of these are standard tracking loops already in use in JPL's flight and ground radios. We first described these functions in continuous-time form, which simplifies the exposition and is easier to relate to standard analog loops of the same type. Then we described digitized versions of each of the major functions, as we've implemented them in Matlab software. We've built two useful software tools from these functional blocks, one allowing the user to simulate the end-to-end operation of a telemetry ranging system, and the other allowing the software to directly process data recorded from a telemetry ranging system's spacecraft and ground receivers. Finally, we derived Cramér-Rao bounds to characterize the accuracy of the various tracking loops, and compared these limits to numerical results obtained from running the software simulation tool.

References

[1] J. Hamkins, P. Kinman, H. Xie, V. Vilnrotter, and S. Dolinar, "Telemetry ranging: Concepts," *The Interplanetary Network Progress Report*, vol. 42-203, Jet Propulsion Laboratory, Pasadena, California, pp. 1–21, November 15, 2015.
http://ipnpr.jpl.nasa.gov/progress_report/42-203/203C.pdf

[2] P. Kinman, "Pseudo-Noise and Regenerative Ranging," *DSN Telecommunications Link Design Handbook*, DSN No. 810-005, module 214, Rev. A, Jet Propulsion Laboratory, Pasadena, California, October 28, 2015.
http://deepspace.jpl.nasa.gov/dsndocs/810-005/214/214A.pdf

[3] C. DeBoy, C. Haskins, T. Brown, R. Schulze, M. Bernacik, J. Jensen, W. Millard, D. Duven, and S. Hill, "The RF telecommunications system for the New Horizons mission to Pluto," *Proceedings of the IEEE Aerospace Conference*, vol. 3, p. 1478, March 2004.

[4] CCSDS 414.1-B-2, "Pseudo-noise (PN) Ranging systems," Blue Book, Issue 2, Mar. 2014.
http://public.ccsds.org/publications/archive/414x1b2.pdf

[5] K. Andrews, J. Hamkins, S. Shambayati, and V. Vilnrotter, "Telemetry-based ranging," in *Proc. IEEE Aerospace Conference*, March 2010, pp. 1–16.

[6] J. Barry, *Digital Communication*, Kluwer Academic Publishers, Boston, 2004.

[7] J. G. Proakis, *Digital Communications*, McGraw Hill, Inc., New York, NY, fourth edition, 2001.

[8] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, McGraw Hill, Cambridge, Massachusetts, 1990.

[9] CCSDS 131.0-B-2, "TM synchronization and channel coding," Blue Book, Issue 2, Aug. 2011.
http://public.ccsds.org/publications/archive/131x0b2ec1.pdf

[10] S. A. Stephens and J. B. Thomas, "Controlled-root formulation for digital phase-locked loops," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 1, pp. 78–95, 1995.

[11] V. Vilnrotter, J. Hamkins, H. Xie, and S. Ashrafi, "Performance analysis of digital tracking loops for telemetry ranging applications," *The Interplanetary Network Progress Report*, vol. 42-202, Jet Propulsion Laboratory, Pasadena, California, pp. 1–27, August 15, 2015. http://ipnpr.jpl.nasa.gov/progress_report/42-202/202A.pdf

[12] P. Kinman, "34-m and 70-m Telemetry Reception," *DSN Telecommunications Link Design Handbook*, DSN No. 810-005, module 207, Rev. A, Jet Propulsion Laboratory, Pasadena, California, June 13, 2003.
http://deepspace.jpl.nasa.gov/dsndocs/810-005/207/207A.pdf